Joe Lametta

KRIPTONITE

Fuga dal controllo globale.
Crittografia, anonimato e privacy
nelle reti telematiche

NAUTILUS

JOE LAMETTA

KRIPTONITE

FUGA DAL CONTROLLO GLOBALE. CRITTOGRAFIA, ANONIMATO E PRIVACY NELLE RETI TELEMATICHE

NAUTILUS

Arma. L'azione e le cui il loro autonomia e Brigg. A.G. sono di tutti il
più di impianti ENEL e con la cui solo in prima cosa si è da parte di
più in parola, si è all'origine, il reato da diverse A.G. non di azione
di fare il cui solo in via di per tutti il 30.09.1994, e legati a
seguire e il più di matrice anarchica, e quelli di fronte delle armi è
il 1976 e 274/92 e con quella di per i fatti di una pluralità di cui i
rapporti tra il noto Tessitore il Buffini e il più di Orvieto per
Tessitore e di quanto le attività criminali come da parte di Rovereto e
il reato di più che in "gruppi di cui la cattura con il più in danno di
Roma, e "C.O.L.P" e di una serie di particolare interesse e per non di
condanna di non può ipotizzare il cui la data di sera fino a diverse
A.G., per le iniziative di tutti il 30.09.1974, e non ha quindi di
"gruppi di alcuni di Roma, e da "gruppi di cui la cattura e con lui in
modo di Orvieto per le cui il 16.

Convenzioni tipografiche

Utilizzato per	***Stile***
Ciò che appare sullo schermo del computer è evidenziato da un carattere non proporzionale.	`il piano per la fabbricazione della kriptonite e' dietro il quadro in salotto`
All'interno del testo, i nomi dei comandi e le opzioni della riga di comando sono indicati tra parentesi angolari. In *corsivo* opzioni e parametri.	<pgp *nomefile*>
I termini introdotti per la prima volta appaiono sempre in corsivo.	*steganografia*

Sommario

Beh, amico, facciamoci un altro po' di questo whisky, dai. Naaa, non pensarci nemmeno a offrire tu. Sono in grana ormai. Così in grana che nemmeno te lo immagini.

E scommetto che nemmeno ti immagini che questa sbronza potrai raccontarla ai tuoi nipoti prima o poi. È con Joe Lametta che stai a bere, uomo. Rammentalo questo nome, domani lo leggerai su tutti i giornali. Ma non dirlo in giro sino a domani, potrebbe risultare poco igienico, lo sai no? ... Tranquillo, rimettiti seduto a bere, mica c'è bisogno che ti innervosisci. Stanotte parto e chi s'è visto s'è visto. Basta solo un bel passaporto blu con l'aquila americana sopra e ti aprono le porte dappertutto. Specie se il passaporto è autentico. Per Joe Lametta niente più robetta da fasulli pidocchiosi che si fanno fregare alla dogana. Con un bel nome nuovo sopra, un nome da signore, di quelli adatti a un miliardario in vacanza. Li paga poco i suoi impiegati lo zio Sam di questi tempi, sai? Bravo, vedo che capisci anche tu quello che voglio dire. E il mio nome nuovo non lo vengo certo a dire a te, non preoccuparti. E nemmeno ti racconto la mia storia perché sono sbronzo, cosa credi. Per sbronzare Joe Lametta, un'intera cisterna dei pompieri piena di questo piscio di vacca non basterebbe mica. Te la racconto perché mi sei simpatico, ecco tutto. E perché a qualcuno la devo raccontare prima o poi, sennò scoppio. Ma acqua in bocca sino a domani, rammenta. Di gente simpatica con la bocca troppo larga sono pieni i cimiteri. E qualche cliente al becchino l'ho portato pure io...

Perché mi chiamano Joe Lametta dici? Beh è una storia lunga. Da ragazzo mi prendevano in giro perché non avevo un cent in tasca e conservavo una lametta da barba anche sei mesi per risparmiare. I furbacchioni del quartiere, già. Me l'hanno dato loro il nome. E come ridevano. Ma alla fine loro sono sempre lì come mammalucchi a dar fiato alla bocca e io invece con la faccia sempre ben rasata e il vestito sempre ben stirato, la mia strada l'ho fatta. E un bel po' di strada, credimi. "La presenza innanzitutto, Joe". Me lo diceva sempre la mia nonna, è lei che mi ha tirato su, benedetta la sua anima. "Lavato sbarbato e ben vestito sempre, rammenta. Se hai meno di un dollaro in tasca, spendi il tuo ultimo

cent per farti la barba. Ragazze, whisky e tutto il resto dopo. È la bella presenza che distingue un ragazzo di talento come te da un ladro di polli, non lo scordare mai". E io mica l'ho scordato. E con quella lametta ho imparato presto anche a farci qualcosa in più che il contropelo. E quei furbacchioni che mi ridevano alle spalle hanno imparato che ridermi in faccia non conveniva. Già. Da quando Pappas il greco dovette farsi dare quei quindici punti su tutte e due le guance, credo... Per quello sono dovuto scappare dal Bronx. Jella maledetta, ho pensato allora. Ma poi è finita che non è stata mica jella, è stata la mia fortuna invece. Perché alla fine sono piovuto qui a Metropolis. Duro lo ero già per conto mio. Ma furbo ancora mica tanto, direi. A farmi furbo invece l'ho imparato qua. Se lavori con Lex Luthor ti conviene. Lui di duri davvero duri ne compra dodici dozzine al giorno se gli gira. Di grana ne ha. Più di tutti. Ma se non sei furbo, con lui duri poco. Alla prima stupidaggine, due bei scarponi di cemento armato ai piedi e via nel fiume. Ma ho imparato in fretta io, e sono andato avanti, sino a diventare un dirigente a modo mio. Vicepresidente alla rimozione e ai lavoretti sporchi, già. Potrebbero scriverci questo sul cartellino da mettere sulla scrivania, se ce l'avessi. Ma invece ho la mia Colt, niente segretaria ma tutte le ragazze che voglio. E quando ho da parlare a Lex non devo fare altro che chiamarlo personalmente. Filo diretto mi capisci? Mica siamo in tanti a poterlo fare, sai? E ancora meno a poterlo fare senza tremarella. Perché il vecchio Lex con quella sua aria da signore è il più duro di tutti i duri che ho incontrato, niente da dire. Non ha bisogno di dirtelo nemmeno, lo capisci da te. E se non lo capisci da te mica te lo sta a spiegare, sai? Perché lui ordina, gentilmente. Non minaccia mai. Solo, se non stai attento, prima o poi fai una visitina a quel calzolaio che lavora col cemento. E poi dentro quel suo testone calvo ne ha di idee. Una miniera. Hanno un bello strillare i giornali e la TV: "Genio del crimine".... Già, ma lui un genio lo è davvero, finora nessuno è mai riuscito ad incastrarlo. Ma qua a Metropolis è dura lo stesso, amico. Anche per lui a volte. Perché Superman, quando non era impegnato a spupazzarsi Lois Lane o quel finocchietto del suo amico giornalista, o magari tutti e due assieme, dava del filo da torcere pure al vecchio Lex..

Cosa dici? Che si sente in giro che il nostro Amico d'acciaio è incazzato nero? Che non ha più il coraggio di metter fuori il suo faccione? Che la Federal Bank di Metropolis ha dovuto chiudere i battenti e non si sa quando riapre? Che la sua fighetta giornalista ha giurato che piuttosto che dargliela ancora si fa monaca e che il SuperUomo dovrà tirare avanti a SuperSeghe per un pezzo? Mh-mh.. l'ho sentito dire anch'io... Se ne so qualcosa, dici? Ehi amico, sei uno sveglio tu, eh? OK. Io non ne so nulla, chiaramente. Ma se invece ne sapessi qualcosa potrei dirti che è cominciato tutto in una fogna. Non importa dove, le fogne di Metropolis sono più grandi di New York e più nere del buco del culo di un negro. E puzzolenti anche. Più puzzolenti del cesso di questo bar il giorno di San Patrizio, dopo che un'orda di irlandesi ci ha vomitato e pisciato dentro tutta quella birra verde. E non è una puzza da poco quella, credimi. E con tutti quei branchi di ratti grassi come vitelli che ci scorrazzano è l'ideale. Nessuno ti può beccare laggiù. Nemmeno Superman. SuperVista, SuperUdito, SuperOdorato, può andare a farseli benedire dal parroco della cattedrale per quello che gli servono là sotto. Per questo siamo andati così giù nelle fogne. Roba da

prendersi il colera, dico io. Ma stavolta ne valeva la pena. Con me c'erano anche un paio di ragazzi. Bravi ragazzi, molti muscoli e poco cervello, ma tant'è. Al e Louie si chiamavano. E se dico ad Al e Louie di portare una cassa, amico, loro la portano dove dico io, e senza fare troppe domande. Già, una cassa piccola ma pesante, molto pesante... loro non lo sapevano, ma credici o no, c'era la B maiuscola in quella cassa, uomo. La B del Big Bang. Proprio la grande Bomba, nientemeno. Strano sai, uno non ci crederebbe che una bomba termonucleare che può incenerire d'un botto tutta questa fottuta città abbia un'aria così tranquilla... Una scatola di metallo con tre lucette sopra, pesante ma nemmeno troppo grande. Tutto lì. Se avevo paura? Nooo. Tutta una questione di scala. Se sei abituato a trafficare con la nitroglicerina cosa ti frega? Bomba nucleare o cassa di nitro di te non resta neanche un'unghia in nessun caso... Bum, pof. Tutto lì... No amico, niente assalti a basi USAAF, per procurarci la bomba, molto più facile di così, credo. Lo sai cos'è Internet, no? Ne parlano tutti da un po' di tempo a questa parte. E il vecchio Lex vuole che anche noi della vecchia scuola ci capiamo qualcosa in quella roba, anche da prima che se ne parlasse alla TV. "Vi serve" diceva. "Vi tiene al passo con i tempi". Eh, hai ragione, amico, nemmeno a me piaceva tanto quella roba. Computer, modem, tutte quelle stronzate elettroniche. Cose da studentelli con la maglietta del college, mica roba da uomini. E fosse per me non ci avrei pensato mai. Giochetti scemi per ragazzini segaioli coi foruncoli e gli occhiali, ecco quel che pensavo. Ma quando Lex ti dice che faresti bene a fare una cosa, tu mica stai a discutere se ti garba o no. La fai e basta, mi capisci? E anche stavolta ha avuto ragione lui, perché ci è servita, e molto, poi te lo spiegherò. Ma intanto penso che anche i piani per la grande B se li è trovati in quel modo, su Internet... Quanto al cobalto o come diavolo si chiama, il vecchio Lex ha le mani molto lunghe, nemmeno immagini quanto, uomo. E nemmeno io lo voglio immaginare. Lo Zio Sam paga poco i suoi impiegati, te l'ho detto. E anche governatori presidenti e generali a quattro stelle, almeno a sentire loro. Hai ragione amico, tutta questione di punti di vista, ma ci sono cose che è meglio non sapere, se non vuoi fare quella famosa visitina dal calzolaio di Lex, mi spiego? E io c'ho già il mio di calzolaio, e ci sono affezionato, le vedi queste scarpe? Sono italiane, 200 dollari mi sono costate, e con lo sconto.... Su, facciamoci un altro goccio, che ora ti spiego.

Insomma. Noi lasciamo la Bomba lì, al sicuro, dove nemmeno Superman la poteva trovare e ce la filiamo. Lavoretto pulito. I giovanotti tutti muscoli li tengo con me, non si sa mai. Perché capisci, il detonatore della Bomba, il pulsante per il Big Bang ce l'avevo IO, mica Lex. Lui se n'era andato alle Bahamas nel frattempo. A spiegare al Sindaco che se non mollava un miliardo di dollari in contanti tutta Metropolis gli si vaporizzava sotto il culo ci hanno pensato altri ragazzi, che nemmeno lo sapevano di lavorare per Lex Luthor, pensa un po'... Nemmeno Al e Louie lo sapevano, pensavano di lavorare per me. Quando pensavano, che poi non è che gli capita troppo spesso. E all'altra Bomba, quella scoppiata nel deserto del Mojiave giusto per far vedere che non era un bluff, ci hanno pensato altri tizi ancora. Ah già... Alla storiella del terremoto e dell'epidemia raccontati dal Governo ci avevi creduto pure te, vero? E pure che la Guardia Nazionale era mobilitata per quello, no? .... Vi hanno presi in giro dici? Potevano sfollarvi? Ghghghgh... voialtri patrioti che il

buon vecchio Zio Sam non è Babbo Natale non lo volete mai capire, vero? È per questo che finite sempre con le pezze al culo o stesi in una bella bara con le stelle & strisce sopra... Su, su, ora lascia perdere tuo fratello spappolato in Iraq da un obice saudita, beviti un altro whisky, e vedrai che la malinconia ti passa... Comunque, io stavo lì col mio detonatore e con Al e Louie, in questo bel cottage, a distanza di sicurezza. MOLTO di sicurezza, alla pelle ci tengo io. E Lex alle Bahamas. Solo io e lui sapevamo dove stava la bomba. E sapevamo che l'Omino Coglione d'Acciaio e tutta la sua combriccola avrebbero cominciato a romperci le uova nel paniere appena il Sindaco si sarebbe convinto che non stavamo scherzando. E io dovevo parlare con Lex capisci? Allora, avevo con me 'sto computer portatile. Pronto per la connessione in rete, modem, altri aggeggi... tutto sistemato in una valigetta. Un gioiellino. Me l'aveva consegnato tre giorni prima un tizio, un tipo strano dall'aria sempre sballata che sembra più un cane che un uomo. Lavora pure lui per Lex, solo che nessuno lo sa. Quando ha sentito come mi chiamavo ha fatto una faccia buffa. Come se ridesse sotto quei baffi radi che c'ha. "Joe Lamerz", sghignazzava, "Joe Lama", o qualcosa del genere... Bah. Non è igienico litigare con un altro che lavora per Lex prima di un colpo come quello, anche se è un fregnetto sballato che dice sempre "yuk yuk". Altrimenti quei dentoni che si ritrova mica li riportava a casa sani, te lo dico io. Lama a me? I lama sono bestiacce che sputano e puzzano, li ho visti allo Zoo una volta... E il qui presente Joe Lametta non sputa mai, quasi mai per lo meno, e spende 75 dollari al mese in acqua di colonia... Comunque, avevo con me questo computer, e l'ordine di accenderlo al cottage dopo aver piazzato la Bomba. Avrei trovato lì le istruzioni per tenermi in contatto con il capo. Per farla breve, lo apro, lo accendo, e il primo file che mi si apre automaticamente è:

Introduzione

di Luc Pac

Libertà e controllo sociale. Comunicazione, tecnologia, dominio, responsabilità. Su questi temi abbiamo qualcosa da dire, qualcosa di diverso dagli equilibrismi teorici. Queste sono infatti pagine molto *pratiche*, che vogliono aprire una nuova strada su un terreno finora dominato da fatalismi ideologici, appelli democratici e richiami a nuove morali universali. Dal canto suo questo libro è probabilmente qualcosa di molto a-morale, nel senso che gli strumenti qui offerti possono essere messi al servizio di etiche personali molto diverse tra loro. Ma la possibilità che questo libro possa finire "nelle mani sbagliate" non rientra comunque tra le nostre preoccupazioni: preferiamo considerarlo piuttosto un problema in più per tutti coloro che sono soliti fare appello al senso di responsabilità degli individui solo quando per qualche motivo gli altri (più "convincenti") metodi di controllo non funzionano più.

Nonostante la sua immediata concretezza e nonostante gli argomenti trattati siano piuttosto complessi, abbiamo cercato di mettere assieme qualcosa che potesse essere letto su più livelli, mantenendo motivi di interesse anche per chi non ha mai utilizzato un personal computer, ma allo stesso tempo senza annoiare i lettori tecnicamente più esperti.

Prima di affrontare direttamente, nei prossimi capitoli, l'utilizzo delle varie tecniche, è importante delineare il contesto sociale e politico in cui ci muoviamo. Per questo motivo le prossime pagine provvederanno a sfiorare diverse questioni, apparentemente eterogenee, importanti per capire quali sono le parti e i fattori in gioco su questo terreno.

Pacchetti e circuiti

Si sprecano ormai le leggende, i libri e gli articoli sulle origini "underground" di Internet. Non vogliamo convincere nessuno che Internet sia davvero un organismo "anarchico" come spesso si racconta - ognuno è libero di verificare e farsi una propria opinione in proposito; quello che vorremmo fare piuttosto è richiamare l'attenzione su un paio di concetti utili anche al fine di capire meglio le prossime pagine di questo libro, che proveranno a esporre i tentativi di controllo e limitazione delle libertà individuali nell'era telematica e le corrispondenti possibili strategie di autodifesa.

Anzitutto, il concetto di *commutazione di pacchetto* (*packet switching*). Per capirlo, pensiamo prima alla comunicazione telefonica: alziamo la cornetta e componiamo un numero; la centrale telefonica si mette in contatto con altre centrali e dopo pochi secondi possiamo parlare con il nostro interlocutore. In questo caso si crea una connessione continua tra noi e la persona con cui intendiamo parlare. Una volta questa connessione era di tipo sostanzialmente fisico: il cavo che partiva dal nostro apparecchio telefonico veniva collegato all'apparecchio telefonico del nostro interlocutore, attraverso una serie più o meno lunga di raccordi intermedi (effettuati talvolta a mano dalle centraliniste, che staccavano e attaccavano vari spinotti da una specie di gigantesco *rack*). Oggi questa connessione fisica viene emulata via software dalle centrali telefoniche digitali attuali, ma il principio rimane lo stesso. Questo modo di effettuare i collegamenti, stabilendo connessioni dirette e stabili tra due punti della rete, è tipico del mondo della telefonia e viene indicato con il nome di *commutazione di circuito*. La connessione di solito è comoda e veloce, ma presenta alcuni inconvenienti: il più rischioso è che se subentra un guasto lungo la connessione (lungo il *circuito*) questa cade irreparabilmente. Per bloccare una telefonata in corso, in altre parole, è idealmente sufficiente bloccare il circuito in uno qualsiasi dei suoi punti.

Il protocollo di base di Internet, l'IP (Internet Protocol, appunto), non è costruito sul modello telefonico basato sulla commutazione di circuito. Il progenitore dell'IP è stato ideato negli anni '60 per conto dei militari dell'ARPA (Advanced Research Project Agency) preoccupati tra le altre cose, in piena guerra fredda, di costruire un'infrastruttura comunicativa in grado di funzionare anche se i sovietici fossero riusciti a distruggere una o più delle sue "centrali". Questo fatto, ormai largamente e a volte esageratamente riportato un po' dappertutto a mo' di aneddoto, ha condotto a costruire la rete basandosi sul concetto di *commutazione di pacchetto*: non si crea nessuna connessione fissa e stabile tra due punti A e B della rete che vogliono comunicare tra loro; al contrario, il contenuto della comunicazione viene diviso all'ori-

gine in "pacchetti" dotati ognuno di una specie di "intelligenza locale" circa il punto di destinazione. Questi pacchetti vengono sguinzagliati sulla rete e la attraversano in modo indipendente l'uno dall'altro, scegliendo di volta in volta la strada da percorrere in base alle condizioni della rete in quel momento. I due punti A e B non sono quindi uniti da una e una sola linea, bensì da un numero potenzialmente infinito di percorsi che mutano continuamente.

In generale dunque, il controllo dei flussi comunicativi in una rete basata sulla commutazione di pacchetto è qualcosa di molto più complesso del semplice controllo su una centrale sufficiente per bloccare o controllare una comunicazione telefonica.

Il secondo concetto su cui si vuole richiamare l'attenzione è strettamente legato al primo e riguarda la questione del *decentramento*.

Si dice spesso che Internet "non ha padroni". Questo può essere vero o falso a seconda dei punti di vista ed equivale più o meno a dire che "il mondo non ha padroni". Quello che ci interessa di più è che non esiste un vero e proprio organismo centrale di controllo su Internet. Esistono, da una parte, organismi con funzioni consultive e di coordinamento (come l'Internet Society e la sua Internet Engineering Task Force, o come l'italiano GARR); dall'altra parte esistono i singoli Stati con le proprie legislazioni, perlopiù (ma ancora per poco) prive di leggi specifiche sul mondo telematico. La novità è rappresentata dal fatto che, mentre nel mondo "fisico" i confini della giurisdizione statale sono delimitati in modo piuttosto preciso, nel mondo telematico essi sono molto più sfumati: ad esempio, alcuni degli autori di questo libro, anagraficamente cittadini italiani, hanno il proprio recapito telematico presso computer situati all'estero, attraverso i quali ricevono quotidianamente file e corrispondenza varia. Com'è ovvio, essi potrebbero tessere la loro rete di relazioni sociali (fatta di relazioni amorose, contratti commerciali, interscambi culturali o progetti terroristici) con cittadini americani dotati di indirizzi telematici asiatici e via di questo passo.

Nonostante i progetti di integrazione e omologazione sotto il "nuovo ordine mondiale", gli Stati-Nazione attuali (e le loro polizie) sono messi in difficoltà dalla natura intrinsecamente transnazionale dei flussi di comunicazione di qualunque tipo, compresi quelli telematici.

UNDERGROUND

Abbiamo parlato finora di Internet, commutazione di pacchetto e decentramento. Ma paradossalmente le caratteristiche "libertarie" della rete delle reti hanno potuto essere colte, per lunghi anni, solo da elités molto ristrette: nata come si è detto con finalità indirettamente militari, Internet si è evoluta fino ai primi anni '90 quasi esclusivamente all'interno delle università. L'underground telematico viaggiava su altre strade - a parte le sistematiche, ma relativamente limitate, esperienze di hacking extra-legale condotte sulle reti a commutazione di pacchetto più in voga a quei tempi (più che Internet, le vere scuole guida degli hacker erano le reti X.25 e quindi, per l'Italia, Itapac). Se si prescinde dunque dalle attività di hacking e pirataggio "in progress", gli esperimenti più interessanti di appropriazione della telematica da parte delle minoranze sociali sono stati condotti (e in parte lo sono ancora) su reti commutate totalmente autogestite. Detto più semplicemente: reti "povere", basate - anche nella loro ossatura centrale - su modem e semplici linee telefoniche (dove Internet utilizza invece costose e veloci linee dedicate, spesso su fibra ottica), ma con la notevole caratteristica di essere totalmente inventate dal basso e prive di qualsiasi legame con università o altre istituzioni. L'assoluta libertà e indipendenza di queste reti si manifestano su due versanti: da una parte la possibilità di entrare a far parte della rete come "nodo" (e quindi fornitore di servizi) a tutti gli effetti, dall'altra quella di partecipare alla vita di rete come semplice utente. In entrambi i casi l'attrezzatura tecnica necessaria è limitata a personal computer, modem e normale linea telefonica. Tutto il resto avviene tramite accordi e regole perlopiù informali che gli stessi utilizzatori della rete contribuiscono a stabilire. Le dimensioni ristrette e spesso geograficamente delimitate hanno portato alla proliferazione di reti autogestite (in cui l'elemento di base è costituito dalle BBS locali) dedicate agli argomenti più vari: ormai storiche ad esempio quelle orientate alle minoranze sessuali.

La più grande, la più famosa e forse anche la più longeva di queste reti è Fidonet. Creata nel 1984 da Tom Jennings, egli stesso dichiaratamente "hacker punk, omosessuale e anarchico" con lo scopo di facilitare la comunicazione tra soggetti in qualche modo "affini", si è poi involuta negli anni per presentarsi, soprattutto nella sua "sezione" italiana, come un macchinoso apparato di regole e cavilli formali, moderatori/censori e altre figure gerarchiche. Forte di decine di migliaia di nodi in tutto il mondo e di centinaia di conferenze elettroniche, Fidonet continua tuttavia ancora oggi a essere un'alternativa al mondo di Internet. E proprio da un interstizio "deviante" di Fidonet nasce, nel 1993, la rete Cybernet, circuito "liberato" italiano destinato a ospitare discussioni e progetti su cyberpunk, nocopyright, hacking, phreaking,

crittografia d'assalto - ma anche outing sessuale, droghe, musica elettronica e altri argomenti poco battuti negli ambienti mainstream.

A Cybernet si affianca la parte italiana di ECN (European Counter Network), rete di stampo più tradizionalmente politico diffusa soprattutto all'interno di centri sociali autogestiti e collettivi della sinistra autorganizzata. Nonostante alcune differenze di base che continueranno a riemergere periodicamente, Cybernet ed ECN vanno a costituire un unico spazio telematico atipico e in qualche modo sommerso, sia rispetto alla più "presentabile" Fidonet, sia rispetto ai nuovi fasti di Internet.

Il motivo di questa breve digressione sulla storia della telematica *underground* in Italia è dato dalla duplice importanza che le reti Cybernet/ECN, e in particolare la conferenza *cyberpunk*, rivestono nell'ideazione di questo libro. In primo luogo esse hanno costituito l'*humus*, la casa telematica, il luogo sociale in cui i vari autori di queste pagine si sono incontrati e hanno maturato il loro interesse e la loro competenza sui temi della tutela della sfera individuale nella società digitale. Proprio nella conferenza cyberpunk, ad esempio, un messaggio del 12 agosto 1993 introduceva pubblicamente, forse per la prima volta in Italia, il dibattito sull'utilizzo del software *Pretty Good Privacy* (PGP), di cui si parlerà in seguito.

In secondo luogo, in tempi precedenti l'attuale "caccia al pedofilo in rete", Cybernet/ECN hanno rivestito per la prima volta il ruolo del "cattivo telematico" nell'immaginario collettivo di sbirri, magistrati e giornalisti italiani. Da diversi anni i riferimenti a queste reti compaiono in modo più o meno aperto e marcato nei rapporti periodici sui pericoli di eversione presentati dai servizi segreti al Parlamento. Proprio in riferimento alle reti Cybernet/ECN, nella 33a "Relazione sulla politica informativa e della sicurezza", relativa al 1° semestre 1994, presentata dalla Presidenza del Consiglio dei Ministri al Parlamento, si legge ad esempio che "è stato seguito con attenzione l'interesse dei gruppi antagonisti all'impiego di reti telematiche per la raccolta e la diffusione di notizie di area nonché alla potenzialità, in chiave antistatale, degli strumenti informatici. A quest'ultimo riguardo non sono stati sottovalutati gli aspetti di pericolosità connessi all'eventuale sviluppo di tali tecnologie per introdursi illegalmente in archivi pubblici e privati e acquisire informazioni riservate, la cui divulgazione potrebbe avere ripercussioni negative per la sicurezza". La preoccupazione repressiva per ciò che accade in rete risale quindi a qualche anno fa e ha radici ben diverse dal perbenismo moralista con cui la si vuole giustificare oggi.

Piccoli mostri

I professionisti del controllo sociale si sono accorti abbastanza presto che, con la massiccia introduzione dell'alta tecnologia nella società, non tutto stava andando per il verso giusto. Certo, in una qualche misura il mondo correva verso il fatidico 1984 di Orwell in cui l'occhio del Potere sarebbe penetrato nelle case di tutti attraverso i teleschermi. Ma se da una parte si stavano effettivamente sviluppando quegli strumenti e quelle tecnologie che oggi permettono, ad esempio, di controllare gli spostamenti di una persona attraverso telecamere fisse, satelliti e telefoni cellulari usati come microspie ambientali o localizzatori di posizione - dall'altra parte si intravedeva la forma di alcuni "piccoli mostri" che avrebbero ben presto mostrato al mondo intero le nuove contraddizioni e le debolezze di una società basata sull'informazione.

Negli anni '80 partono infatti le prime paranoie e i primi processi contro il famigerato *pericolo hacker*. Sulla scia del film *Wargames*, addetti alla sicurezza, uomini politici, poliziotti, giornalisti, insegnanti, genitori e soprattutto ragazzini svegli ma annoiati dalle carceri scolastiche, si rendono conto che la società americana, la più tecnologicamente avanzata al mondo, sta fidandosi un po' troppo delle macchine.

Le macchine non sono solo strumenti di controllo sociale; il loro uso può essere distorto e piegato alle necessità individuali: quello che è necessario fare in ogni caso è "metterci le mani sopra". Questo era il messaggio degli hacker, messaggio nato in realtà alla fine degli anni '50 nei laboratori universitari del MIT, ma rimasto ascoltato da pochi fino al momento della diffusione di massa dei personal computer. Uno dei tanti a fomentare questa voglia di "mettere le mani sopra" a sistemi fino ad allora considerati magici e inavvicinabili fu John Draper, alias *Captain Crunch*, col suo fischietto a 2600 Hertz capace di far impazzire i contascatti delle centrali telefoniche. L'arte del *phreaking* e dei vari metodi per telefonare senza pagare costituisce un esempio di uso creativo della tecnologia per la soddisfazione "unilaterale" di un bisogno primario, quello di comunicare con i propri simili. "Unilaterale" in quanto non passa attraverso forme organizzate di rappresentanza degli interessi. Nessun rappresentante al Congresso o al Parlamento, insomma, nessuna proposta o controproposta di legge, solo i singoli phreaker e i loro marchingegni capaci di realizzare *qui e ora* i propri desideri.

Il "pericolo hacker" amplificato dai media ha portato nelle aule dei tribunali numerose vittime. Impossibile e inutile elencarle tutte, ci limitiamo a un caso eccellente.

Nel 1990 avviene negli USA l'operazione Sun Devil, la prima azione repressiva pubblica e su vasta scala nei confronti degli hacker. Tra gli imputati, Creig Neidorf, meglio conosciuto in rete con lo pseudonimo di Knight Lightning, editor della rivista elettronica Phrack. È accusato dai servizi segreti di aver pubblicato sulla sua rivista un documento riservato sul funzionamento dei servizi telefonici di emergenza americani.

Ovviamente non ci interessa dimostrare, come hanno invece cercato di fare i suoi legali, che Knight Lightning fosse in realtà un bravo cittadino americano solo un po' troppo curioso. Quello che ci interessa è piuttosto il fatto emerso dal processo (e che, tra l'altro, ha determinato il proscioglimento dalle accuse di Neidorf): il documento "riservato" incriminato, il file segreto sui sistemi telefonici 911 che sarebbe stato trafugato con sofisticate tecniche di hacking dai computer dell'AT&T (la compagnia telefonica americana) faceva parte in realtà del materiale informativo/promozionale che la stessa AT&T inviava a casa per corrispondenza per soli 5 dollari a chiunque ne facesse richiesta.

Qualcuno inizierà a domandarsi cosa c'entra questa lunga divagazione sull'underground telematico, gli hacker e i pirati, con il tema centrale di questo libro. Ma proprio i primi casi esemplari di repressione contro gli hacker mostrano quanto le agenzie preposte al controllo sociale abbiano *paura* di chi si appropria direttamente di determinate conoscenze. La società digitale, tanto decantata in negativo anche da molte voci "di sinistra" o "anarchiche" come un qualcosa di assolutamente monolitico, centralizzato, pervasivo, in cui lo spazio concesso all'autonomia individuale si sarebbe annullato, ebbene questa società digitale fa acqua da tutte le parti, e gli hacker l'hanno dimostrato. Ciò che terrorizza sbirri, giudici e politici è proprio l'atteggiamento *hands on* degli hacker, l'atteggiamento di chi intende "mettere le mani sopra" le macchine, di chi sfrutta a proprio piacimento i terrificanti "buchi" nella sicurezza delle reti telematiche e le clamorose contraddizioni di una società che vorrebbe applicare le sue vecchie leggi e i suoi strumenti repressivi a qualcosa di nuovo e sfuggente come l'informazione digitale. Di più ancora, fa paura l'atteggiamento "unilaterale" degli hacker che non riconoscono nei partiti politici, nel governo o nello Stato alcuna controparte con cui mediare. Proprio questo atteggiamento sarà l'eredità che intendiamo raccogliere con questo libro.

Non è superfluo notare anche come nessuna legge abbia mai potuto regolamentare l'hacking nelle sue varie forme: certo, con il passare degli anni ormai quasi tutti i paesi hanno ottenuto le proprie leggi specifiche anti-hacker (in Italia la legge 547 del 23 dicembre 1993, la famosa legge Conso, punisce con la reclusione fino a tre anni l'accesso abusivo ai sistemi telematici

– anche qualora tale accesso non provochi alcun danno). Tuttavia queste leggi poco hanno potuto contro quell'atteggiamento "hands on" che costituisce la vera anima degli hacker e che continua ancora oggi a prescindere da qualunque artificiosa demarcazione tra legale e illegale.

Bibliotecari o editori?

Se il "pericolo hacker" ha monopolizzato l'attenzione dei media fino ai primi anni '90, la successiva esplosione di Internet e la sua commercializzazione presso un pubblico di milioni di utenti hanno portato alla ribalta questioni ben più spinose e complesse delle semplici intrusioni telematiche non autorizzate. L'incapacità (e forse l'impossibilità) di risolvere queste questioni è ciò che sta alimentando l'attuale spostamento dell'attenzione pubblica verso temi di facile presa emotiva, pornografia e pedofilia prima di tutto.

Questi "grandi mostri" dell'immaginario collettivo occidentale (alle "tre P" di pornografia, pedofilia e prostituzione si affiancano i trafficanti di "droga" e talvolta i "terroristi internazionali") servono a creare un alibi e un clima adatti per il tentativo di una svolta liberticida nel modo diffuso di considerare la rete.

Ciò che infatti rimane nascosto dalla cosiddetta "emergenza pedofili" è l'insieme delle sfide portate alla cultura poliziesca da parte di un mondo che - a modo suo - non riconosce più confini nazionali, identità anagrafiche e leggi sulla proprietà, ma solo la prassi della soddisfazione dei propri desideri, sociali o antisociali che siano. Sono queste sfide a rimanere nascoste dalla densa coltre di fumo sollevata dalle crociate anti "pedofili", e sono sfide che pur esplicandosi quotidianamente a un livello sommerso o "underground", riescono a raggiungere talvolta anche una visibilità pubblica nelle aule dei tribunali o tra le righe dei comunicati stampa.

Vorremmo chiarire meglio, attraverso alcuni esempi, quali possono essere queste sfide.

In primo luogo c'è la questione della responsabilità su ciò che viene immesso/comunicato in rete. Non è un caso che tutti gli Stati si siano dotati da tempo di rigidissime regolamentazioni sull'utilizzo dei mass media; in Italia, l'obbligo di avere un direttore responsabile iscritto all'albo (e quindi membro della ristretta corporazione dei giornalisti) per tutte le pubblicazioni

periodiche è un chiaro sintomo della necessità di tenere sotto controllo l'utilizzo di strumenti atti a veicolare idee, opinioni, denunce a un pubblico molto più vasto di quello consentito dai soli contatti interpersonali. Ogni parola detta o scritta "in pubblico" deve avere un responsabile identificato anagraficamente (vale a dire, in base ai registri dello Stato). Gli spiragli lasciati all'anonimato o allo *pseudonimato* sono esigui: le scritte sui muri (anche queste comunque ufficialmente perseguibili), il telefono o i servizi postali (adatti però solo a comunicazioni interpersonali, *one-to-one*). A questo proposito è anche interessante notare la rigidità della regolamentazione vigente sulle comunicazioni che sfruttano le onde radio: tralasciando le stazioni televisive e radiofoniche broadcast, per le quali sono necessarie apparecchiature costose e relativamente sofisticate, l'etere (ne parleremo in uno dei prossimi capitoli) potrebbe costituire uno straordinario *medium* comunicativo economico, aperto e alla portata di tutti. Forse è proprio per questo che l'attività di radioamatore oggi è così strettamente ingabbiata da licenze, permessi e controlli da parte di speciali organi di polizia.

Se ben indirizzato un messaggio in rete può raggiungere migliaia di destinatari, con spese e difficoltà minime. La rete si configura quindi come un nuovo mass medium, in cui la comunicazione può avvenire *many-to-many* e non solamente *one-to-many* come nei media tradizionali. In altre parole, le difficoltà (economiche, organizzative, legali) per divenire *fornitore* di informazioni sono pari a quelle sufficienti per essere semplice *consumatore* delle stesse. Quindi non più pochi, grossi organi di informazione centralizzati, bensì una miriade di megafoni, riviste, bollettini, radio e televisioni (in quanto il traffico in rete è sempre più spinto verso una dimensione multimediale, in primo luogo per evidenti ragioni di *appealing* commerciale) realizzati in casa e spesso dalla vita brevissima.

Se questo ovviamente non ha nulla a che vedere con il problema di *cosa* comunicare (sono in molti a considerare le "informazioni" che girano in rete per il 99% pura spazzatura), è anche vero che la semplice possibilità concessa a chiunque di rivolgersi direttamente a migliaia di altre persone costituisce, appunto, una *sfida* all'arroganza con cui lo Stato ha finora regolamentato il "diritto" di parola.

La questione è dunque la seguente: date le possibilità attuali di inviare un messaggio (un testo, un file audio, un video, un'immagine) a larga diffusione e in modo virtualmente anonimo (e sull'anonimato ci torneremo dettagliatamente), chi è da ritenersi responsabile per le eventuali conseguenze, legali o di altro tipo, del messaggio stesso? Detto in pratica e con un esempio attuale: poniamo che qualcuno utilizzi la rete per diffondere pubblicamente un'ipotetica conversazione telefonica cellulare catturata tramite radioscanner in cui

il giudice romano Antonio Marini confida a un collega la natura "inventata" della sua inchiesta contro il movimento anarchico italiano. Ebbene, nei confronti *di chi* Antonio Marini potrebbe far partire la sua probabile (e probabilmente vincente) denuncia? Considerato che l'autore effettivo del messaggio può, con un minimo di attenzione, rimanere totalmente anonimo, in casi simili realmente successi gli strali repressivi hanno cercato di concentrarsi sull'anello successivo della catena: il fornitore del servizio presso il quale è partito il messaggio.

I fornitori di servizi telematici, siano essi grossi e potenti Internet Provider commerciali oppure semplici gestori di BBS amatoriali, sono quindi al centro di una diatriba legale che in tutto il mondo tenta di farli rientrare all'interno di categorie giuridiche tradizionali e collaudate. In particolare, la questione è la seguente: i fornitori di servizi telematici devono essere considerati come *editori* (quindi responsabili a tutti gli effetti di quanto veicolano) oppure come *bibliotecari* o *edicolanti* (quindi con semplici funzioni di depositari di materiali sui cui contenuti non hanno né possono avere completa conoscenza)? È da ricordare che, a differenza di quanto accade con le pubblicazioni a stampa, i messaggi su una rete telematica transitano in modo principalmente automatico e in tempo reale, dunque senza possibilità di controllo e censura preventiva. Chiedere ai gestori di diventare responsabili, attraverso un controllo preventivo, di ciò che veicolano significherebbe in pratica bloccare la loro attività con ovvie e immediate ripercussioni economiche sul nascente mercato basato sulla comunicazione on-line.

È interessante citare uno dei pochi precedenti legali in proposito, avvenuto su suolo americano. La scarsa giurisprudenza esistente in America tende a considerare, correttamente, gli Internet Provider come "bibliotecari" piuttosto che come "editori", alleggerendoli quindi dalle responsabilità sul contenuto di ciò che veicolano. Una sentenza del tribunale di New York del 24 maggio 1995 (*Stratton Oakmonth v. Prodigy*) ha invece deciso diversamente, ma con una motivazione che di fatto conferma esplicitamente la possibilità per i provider di essere considerati normalmente "bibliotecari" e non "editori". Il caso opponeva una ditta di consulenze finanziarie al servizio telematico Prodigy. Un utente, rimasto anonimo, ha inserito un messaggio considerato diffamatorio in una delle conferenze pubbliche di Prodigy. Quest'ultima è stata ritenuta responsabile e costretta a pagare i relativi danni. La motivazione della sentenza, però, si appella a una particolare caratteristica che distingue Prodigy da quasi tutti gli altri provider: per vendersi sul mercato come servizio adatto alle famiglie (si sa, i genitori apprensivi per le navigazioni dei figli sono sempre di più) Prodigy adotta proprio una forma di controllo preventivo sui messaggi delle proprie conferenze, censurando ciò

che ritiene in qualche modo "poco adatto". Il fatto che questa "caratteristica in più" venisse ampiamente pubblicizzata, secondo il giudice autorizzava gli utenti ad attendersi proprio un servizio "evoluto" simile a quello di un giornale o una rivista. In altre parole il giudice ha considerato Prodigy responsabile dei contenuti di ciò che veicola *proprio perché su Prodigy esiste un tentativo dichiarato di controllo, sia pure parziale, dei messaggi che circolano*. La stessa sentenza ha anche esplicitato che, normalmente, tale controllo non viene effettuato, gli utenti ne sono consapevoli, e quindi i gestori di servizi telematici devono generalmente essere considerati alla stregua di bibliotecari o edicolanti.

Con tutte le cautele del caso, sembrerebbe dunque che un approccio libero, incontrollato e non paranoico alla comunicazione in rete offra le migliori garanzie *perfino* da un punto di vista legale. Questo riconduce al dibattito in corso da vari anni fra le diverse reti telematiche amatoriali anche in Italia, in particolare fra quelle - come Fidonet - propense a regolamentazioni e controlli più o meno rigidi, e quelle - come Cybernet - dichiaratamente prive di qualsiasi controllo sia sul contenuto dei messaggi che sull'identità anagrafica dei mittenti. Non solo, questa prospettiva ha evidenti ripercussioni anche sui metodi di gestione della posta personale. Anche qui, nella telematica amatoriale si assiste da anni al contrapporsi di due filosofie: da una parte i gestori che dichiarano di monitorare periodicamente i messaggi dei propri utenti, con lo scopo di scoraggiare l'utilizzo dei loro sistemi telematici a fini illeciti; in questi casi è di norma anche proibito l'utilizzo di tecniche di crittografia (come il PGP) perché ciò impedirebbe ai gestori di prendere visione dei contenuti dei messaggi. Dall'altra parte i gestori che rinunciano al monitoraggio dei messaggi per motivi di impraticabilità tecnica, e che da parte loro *incoraggiano* l'uso della crittografia in quanto l'impossibilità matematica di prendere visione dei messaggi personali li esonererebbe da qualsiasi responsabilità sul contenuto degli stessi.

ALTRE QUESTIONI APERTE

Se la responsabilità dei gestori è una delle questioni più delicate tra quelle dibattute nell'arena politico-istituzionale, l'intrigo di scontri tra censori statali, imprenditori rampanti liberal-capitalisti, "forze progressiste" e tecnoanarchici incazzati offre molti altri esempi. Ne citiamo ancora qualcuno per renderci conto di come il mondo digitale sia in grado talvolta di mettere in crisi i tradizionali meccanismi di controllo sociale.

1993: Robert e Carleen Thomas, marito e moglie, gestiscono una BBS (un servizio telematico amatoriale) a Milpitas, California. La BBS si chiama Amateur Action BBS e contiene, tra le altre cose, immagini pornografiche dedicate a pratiche particolari (feticismo, coprofilia, *bestialities*). L'accesso alla BBS e in particolare a queste immagini non è immediato per chiunque: è necessario infatti formulare una richiesta formale di accesso, pagare un abbonamento e dichiararsi consapevoli e interessati al contenuto delle immagini. Si noti che, nella libertina California, queste immagini sono considerate perfettamente legali e sono anzi tutelate dal primo emendamento della costituzione americana sulla libertà di espressione. Per quanto alcune di queste immagini possono risultare disgustose per molte persone, coloro che le richiedono sono quindi perfettamente consapevoli del loro contenuto, sono maggiorenni e hanno forse l'unica "colpa" di dedicarsi a pratiche sessuali disapprovate dalle morali più conservatrici secondo gli standard locali.

Per diverso tempo, quindi, tutto fila liscio; fino a quando un solerte investigatore federale si collega ad Amateur Action BBS sotto falso nome, compila il modulo di ingresso e preleva alcune di queste immagini. Sfortunatamente per i coniugi Thomas, l'investigatore in questione ha pensato bene di scegliere la propria base operativa (e quindi il luogo in cui ricevere i file) non in California, dove i Thomas gestiscono la loro BBS, bensì a Memphis, Tennessee. La scelta di agire dal Tennessee, stato del Sud reazionario e conservatore, non è casuale: una decisione della Corte Suprema degli Stati Uniti del 1973 stabilisce che i casi di "oscenità" e offesa alla decenza devono essere giudicati in base *agli standard della comunità locale.*

Risultato: Robert e Carleen Thomas, californiani, operatori di un servizio telematico situato in California, immersi nell'atmosfera e negli "standard" sociali e morali della California, vengono giudicati colpevoli di "oscenità" in base agli "standard" locali del Tennessee, e condannati rispettivamente a 37 e 30 mesi di carcere.

Questa sentenza suscita immediatamente un prevedibile clamore in rete. Ciò che è avvenuto non ha senso: i segnali che corrono lungo i cavi non conoscono confini nazionali e gli utilizzatori abituali della rete sanno bene che gli "standard di decenza" della loro comunità, se mai ve ne sono, sono quelli del ciberspazio. È interessante notare però, in questo caso, che una simile sentenza non ha solamente (e giustamente) indignato i libertari o i paladini della libera espressione, bensì anche tutti i fornitori *commerciali* di servizi telematici: dai sexy shop che vendono i propri articoli attraverso la rete, ai supermercati on-line, alle grosse banche dati. Se un simile precedente giuridico dovesse prendere piede, infatti, magari esteso a livello internazionale, il contenuto di qualsiasi servizio on-line dovrebbe adeguarsi *al più restrittivo*

tra tutti gli "standard morali" conosciuti - ma chiunque abbia anche solo qualche minima nozione di antropologia culturale si rende immediatamente conto che la vastità delle culture e delle morali umane diffuse sul pianeta porterebbe rapidamente ad annullare qualsiasi contenuto, e con esso la possibilità di condurre affari sulla rete. Una prospettiva sicuramente e radicalmente anti-capitalista, che il Mercato non potrebbe mai tollerare. Un bel problema, insomma...

Ma passiamo a un'altra storia.

Premessa: *alt.religion.scientology* è un *newsgroup*, una conferenza elettronica pubblica dedicata alla Chiesa di Scientology. Vi partecipano soprattutto fuoriusciti dalla Chiesa che denunciano le pratiche di adescamento e di lavaggio del cervello subite dagli adepti. Per questo motivo i messaggi postati in questa conferenza vengono tenuti attentamente d'occhio da parte degli uffici legali di questa potentissima setta, che talvolta provvedono a intimidazioni e denunce nei confronti dei loro autori. *Anon.penet.fi*, invece, era un particolare servizio telematico localizzato in Finlandia esistente fino al 1996, in particolare era uno dei primi *anonymous remailer* esistenti su Internet. Questi servizi verranno dettagliatamente spiegati da un punto di vista tecnico in un apposito capitolo di questo libro - per ora ci limitiamo a presentare un *anonymous remailer* come un servizio che consente di inviare messaggi anonimi a un qualsiasi indirizzo di posta elettronica.

Naturalmente, più di un ex-aderente alla Chiesa di Scientology ha pensato bene di sommare le due cose e di sfuggire alle sfiancanti cause legali degli avvocati delle Chiesa, inserendo i propri messaggi in *alt.religion.scientology* attraverso il servizio offerto da *anon.penet.fi*, cioè anonimamente.

Anche qui, lo stratagemma ha funzionato fino a quando i "mastini" di Scientology non sono riusciti a prendere le opportune contromisure tecniche e legali: cioè fino al 22 agosto 1996, quando il tribunale di Helsinki, su pressione dei legali della Chiesa, ha ordinato a Julf Helsingius, operatore del servizio anon.penet.fi, di rivelare il reale indirizzo elettronico (e quindi l'identità) dell'ennesimo critico della setta, che questa volta aveva utilizzato il remailer per postare in pubblico alcuni documenti considerati "testi sacri e segreti" da Scientology. Helsingius, per evitare guai maggiori, ha acconsentito. Ma le vivaci polemiche che hanno accompagnato questa sua decisione, oltre ad alcune ulteriori vicissitudini negative, lo hanno spinto poco dopo a chiudere definitivamente il suo servizio di anonymous remailing. Ovviamente la comunità internazionale degli amanti della privacy individuale ha immediatamente imparato la lezione: le decine di remailers sorti subito dopo (o addirittura prima) la chiusura di anon.penet.fi utilizzano ora software di nuova

concezione. Un intero capitolo di questo libro è dedicato a spiegarne i dettagli tecnici. Per ora, al fine di dare un senso al nostro discorso, ci limitiamo a segnalare che i gestori di questi nuovi remailers, a differenza di Julf Helsingius, non conoscono né possono più tecnicamente conoscere i reali indirizzi elettronici dei propri utilizzatori e quindi, naturalmente, non possono essere tenuti a rivelare alcunché.

TRAFFICO D'ARMI PER TUTTI

> "*Il più antico dei trattati sulla guerra conosciuti, scritto dallo stratega cinese Sun Tzu (ca. 400 a.C.) fa consistere l'essenza del combattimento non nell'esercizio della violenza, bensì nella capacità di prevedere e ingannare, cioè nella preconoscenza necessaria a esprimere valutazioni sull'andamento di una campagna e nei mezzi adatti a ingannare un potenziale nemico riguardo alle proprie inclinazioni e intenzioni reali. A causa del ruolo-chiave svolto dalla conoscenza e dall'inganno nelle questioni militari, gli eserciti dell'antichità (gli eserciti egizio, assiro e greco, per esempio) avevano già sviluppato approcci sistematici per la raccolta e l'analisi delle informazioni, così come per le arti occulte e il controspionaggio.*" (Manuel De Landa, *La guerra nell'era delle macchine intelligenti*, Feltrinelli, 1996, p. 272)

I primi calcolatori elettronici (l'americano ENIAC e il britannico Colossus) furono messi a punto durante la Seconda Guerra Mondiale con compiti specifici di raccolta ed elaborazione di informazioni: il computer ENIAC era dedicato alla ricerca balistica, mentre il Colossus fu progettato con il compito di decrittare il sistema di crittografia utilizzato dal comando strategico nazista per comunicare gli ordini alle truppe (il famoso codice Enigma).

Quando i calcolatori delle "truppe alleate" riuscirono effettivamente a decodificare il codice Enigma, tale successo costituì un vantaggio strategico incolmabile: gli americani furono in grado di conoscere in anticipo le mosse dei tedeschi senza che gli stessi tedeschi, convinti della sicurezza del proprio codice, se ne rendessero conto. Nella storia non scritta della Seconda Guerra Mondiale, fu probabilmente questo il fattore che più contribuì alla sconfitta delle forze tedesco-giapponesi, molto più della bomba di Hiroshima. Non è un caso infatti che a partire dall'immediato dopoguerra gli Stati Uniti costi-

tuirono uno dei servizi d'informazione più segreti e misteriosi che esistano, la NSA (National Security Agency), dedicato interamente allo studio e all'analisi dei sistemi di comunicazione strategici. Le risorse utilizzate dall'NSA vanno da un foltissimo gruppo di esperti linguisti (sempre nel corso della guerra gli americani impiegarono nelle loro comunicazioni perfino un gruppo di indiani Navaho, la cui lingua pare essere una delle più incomprensibili sulla faccia della terra), fino alla più massiccia concentrazione di potenza di calcolo esistente al mondo. I computer dell'NSA, segretissimi e oggetto di molte leggende, si estendono per centinaia di metri quadrati, e il loro unico compito è quello di macinare numeri e algoritmi di crittografia.

Queste note servono a dare almeno una minima idea dell'immensa importanza militare e politica di quella che in apparenza potrebbe sembrare solo una particolare branca della matematica. Riuscire a comunicare in modo che solo gli "amici" capiscano cosa stiamo dicendo può essere decisivo, e naturalmente comprendere le comunicazioni nemiche a loro insaputa può essere altrettanto decisivo. Fino a pochi anni fa, "amici" e "nemici" in crittologia si sono confrontati solo a livello di potenze militari. Oggi, per la prima volta, la possibilità di utilizzare strumenti di crittografia estremamente robusti e sicuri è concessa a chiunque: non solo eserciti nemici ma anche avversari "interni", cospiratori, dissidenti politici, criminali organizzati e amanti lontani. Le polizie di tutto il mondo - e in particolare quelle degli Stati cosiddetti "liberi", che tengono a mantenere una facciata "garantista" nei loro rapporti con la popolazione, sono assolutamente *terrorizzate* da questa possibilità che ostacolerebbe irrimediabilmente la loro attività principale: ficcare il naso nella vita della gente senza farsi scoprire.

Sfortunatamente per loro l'avvento dei personal computer ha offerto esattamente quella potenza di calcolo a basso costo e larga diffusione che era necessaria per mettere a disposizione di tutti algoritmi matematici di crittografia conosciuti da tempo, ma rimasti a lungo inapplicati per scarsità di risorse. Nei primi anni '90 Phil Zimmermann, un americano divenuto poi per qualche tempo figura-simbolo dei criptoanarchici e bestia nera dei servizi segreti, mette a punto il suo software di crittografia *Pretty Good Privacy* (PGP) e lo regala al mondo. Il software funziona su qualsiasi personal computer di fascia medio-bassa, è gratuito e completo di sorgenti (cioè le informazioni necessarie per esplorare minuziosamente il suo funzionamento interno ed eventualmente modificarlo o migliorarlo), secondo una politica di lavoro cooperativo tanto cara agli *hacker* (e tanto sgradita alle grandi software house, Microsoft in testa).

Il PGP viene accolto con enorme interesse, studiato, discusso, sviscerato nei minimi particolari dall'agguerrita comunità internazionale di matematici

e crittografi che lavorano al di fuori degli istituti segreti militari. Il responso unanime è che questo software, alla luce delle attuali conoscenze matematiche, costituisce uno degli strumenti più comodi e sicuri in mano a privati per comunicare in tutta riservatezza. Detto in altre parole, una comunicazione codificata con PGP può essere letta solo dal legittimo destinatario (a patto ovviamente che il software sia stato usato in modo corretto). Se anche i servizi segreti intercettassero il messaggio, con i loro supercomputer avrebbero bisogno di decine o centinaia di anni di calcolo per poterne leggere il contenuto. Per non parlare delle normali forze di polizia.

E se invece di essere un tranquillo cittadino amante della propria privacy, il mittente fosse un pericoloso delinquente o addirittura un *eversore*, questa situazione potrebbe comprensibilmente turbare il sonno di molte persone.

Non è un caso che il PGP (e in generale i software di crittografia "robusta") e i suoi utilizzatori costituiscono ormai da alcuni anni una spina nel fianco di molti governi. Negli USA, anzitutto, NSA e FBI hanno tentato di bloccarne la diffusione in vari modi, causando a Phil Zimmermann noie legali e fastidi personali (come le immancabili perquisizioni negli aeroporti in occasione dei suoi frequenti viaggi all'estero), proponendo nuove leggi sulla crittografia e nuovi standard che permettessero loro di decifrare comunque le comunicazioni in caso di bisogno (come il famigerato Clipper Chip), ma soprattutto appellandosi all'ITAR, l'*International Traffic in Arms Regulations*, il regolamento sul traffico internazionale di armi che negli Stati Uniti disciplina appunto il commercio di armi e munizioni e richiede una speciale licenza e speciali restrizioni alle ditte che vogliano commercializzare con l'estero. In virtù della loro importanza strategico-militare, gli algoritmi di crittografia "robusti" (cioè quelli impenetrabili anche con le risorse di calcolo più avanzate) vengono considerati dall'ITAR alla stregua di *armi da guerra* e la loro esportazione è quindi ufficialmente proibita.

Il PGP è stato sviluppato negli USA, ma si è immediatamente diffuso in tutto il mondo attraverso le reti telematiche. In teoria qualcuno dovrebbe essere punito per questa "esportazione", ma "disgraziatamente" la comunicazione a pacchetto di Internet e la natura digitale di un programma come il PGP non aiutano molto chi vorrebbe applicare alla rete una logica poliziesca. In particolare, un programma informatico non può essere facilmente messo al bando o bruciato come si è usato fare in passato con certi libri. Nonostante questo è curioso notare come molti governi di tutto il mondo, dopo aver preso (giustamente) molto sul serio la minaccia alla propria sovranità causata dalla crittografia personale, stiano conducendo lotte senza speranza per arginare l'uso di questi strumenti da parte dei loro cittadini: oltre alle già citate preoccupazioni dell'FBI negli Stati Uniti, è da ricordare che in paesi

come l'Iran e la vicinissima Francia l'uso di programmi come il PGP è formalmente proibito, e che altri stati europei stanno esaminando nuove proposte legislative in tal senso.

Beh, è tutto molto comprensibile. Con i nuovi sistemi di crittografia si può comunicare via rete, al telefono o anche attraverso la posta tradizionale senza che nessun estraneo possa verificare il contenuto della comunicazione. Gli organismi repressivi e di controllo si trovano improvvisamente impossibilitati a controllare alcunché. Come se non bastasse, gli strumenti per utilizzare questi sistemi sono spesso gratuiti e risiedono in mucchietti di bytes che possono essere riprodotti in infinite copie con minimo sforzo. A questo punto, ciò che rimane da fare a politici, giudici e poliziotti preoccupati per l'ordine pubblico è *proibire*. Anche quando i divieti non hanno più senso, come in questo caso.

A questo proposito, oltre alle parti dedicate espressamente al PGP, tra i capitoli di questo libro se ne troverà uno dedicato alla steganografia: cioè a quell'insieme di tecniche che consentono a due o più persone di comunicare in modo tale da nascondere l'esistenza stessa della comunicazione agli occhi di un eventuale osservatore; ovvero, visto da un altro punto di vista, all'arte che permette a chiunque di usare *tranquillamente* gli strumenti di crittografia, anche dove questi ultimi dovessero essere formalmente proibiti. La crittografia (e dunque la privacy personale nell'era digitale) non può essere proibita, e in questo libro abbiamo il piacere di spiegarne i motivi.

In ogni caso, come per la lunga epopea degli hacker, notiamo anche qui come una legge federale americana (l'ITAR) e diverse leggi nazionali non siano riuscite (né abbiano speranza di riuscire) ad arginare la diffusione di un semplice programma informatico. Il PGP si è diffuso in tutto il mondo *nonostante* la legge e *prima ancora* che i vari gruppi di attivisti potessero organizzarsi per iniziare quell'azione di lobbying politico che in questo momento sta premendo sul Congresso degli Stati Uniti affinché l'esportazione di crittografia robusta rientri nella legalità. Un manipolo di sconosciuti cypherpunk decisi e incazzati ha provveduto a conquistare la propria privacy in modo unilaterale, seguendo la migliore tradizione hacker, fregandosene di leggi, rappresentanti e partiti politici.

Diritti e doveri

Quello che abbiamo dipinto finora è un quadro complesso, con zone di luce e molte ombre, vittorie e sconfitte per ognuna - e sono molte - delle parti in gioco.

Fino a pochi anni fa le reti telematiche costituivano in molti casi una specie di terra franca in cui sperimentare modalità di comunicazione e di esperienza nuove, a volte sciocche o ingenue ma comunque libere di imparare da sé stesse e dai propri errori.

Oggi questa zona franca non esiste più e assistiamo a diversi tentativi di restringere gli spazi di sperimentazione attraverso disposizioni legislative che garantiscano nuovi diritti e assegnino nuovi doveri. La corsa alla regolamentazione del ciberspazio è stata accolta, specialmente da una certa sinistra "illuminata" e progressista, con una serie di espedienti tesi da una parte a ottimizzare in qualche modo la bilancia diritti/doveri e dall'altra a sostenere i disegni di legge "buoni" e a contrastare quelli "cattivi". Assistiamo così al sorgere di associazioni culturali telematiche, a proposte/controproposte/emendamenti legislativi, a campagne e mobilitazioni organizzate a favore della libertà di espressione, come per il *blue ribbon*, e così via.

Proprio il *blue ribbon* costituisce un perfetto esempio di attivismo politico on-line. Dall'inizio del 1996 capita spesso, navigando in rete, di imbattersi in pagine web che mostrano orgogliose la piccola immagine di un nastro blu (*blue ribbon*, appunto) che rimanda a una campagna per la libertà di espressione promossa da varie organizzazioni soprattutto americane. Si tratta di un tentativo di risposta "popolare" a un'iniziativa legislativa liberticida portata avanti dai settori più retrogradi della politica statunitense (essenzialmente cattolici e moralisti). Secondo questi gruppi, una nuova legge, il Communication Decency Act (CDA), avrebbe dovuto impedire la trasmissione su Internet di "comunicazioni indecenti" comprendendo tra queste ultime tutta una serie di argomenti che vanno dai gruppi di discussione gay/lesbici, all'informazione sulle malattie a trasmissione sessuale, alle informazioni sull'aborto. Tutte cose, tra l'altro, di cui si parla tranquillamente anche *al di fuori* della rete. In tutto il mondo, l'immagine del *blue ribbon* esibita sulla propria pagina web ha testimoniato l'adesione alla campagna contro il Communication Decency Act e per la libertà di espressione in rete. Tale campagna è stata promossa come si è detto da alcune organizzazioni per i diritti civili con sede negli Stati Uniti, tra le quali spicca *l'Electronic Frontier Foundation*, fondazione agguerrita sui fronti anti-censura e per il libero commercio finanziata anche, più o meno direttamente, da colossi dell'informatica come Sun Microsystems e Lotus Corporation. Questa campagna per il

free speech, immediatamente diffusasi a macchia d'olio in tutto il mondo, ha in effetti ottenuto alcuni risultati concreti: oltre ad aver raggiunto una generica maggiore consapevolezza sull'importanza della libertà di espressione, nel 1997 il CDA è stato dichiarato incostituzionale dalla Corte Suprema degli Stati Uniti.

Questo tipo di attivismo politico "militante" può quindi essere interessante ed efficace, ma ignora (a volte volutamente) tutte quelle possibili strade che non passano attraverso la rappresentanza, l'associazionismo ufficiale, il riconoscimento e l'accettazione dell'autorità delle istituzioni. Le critiche verso questo variegato arcipelago "progressista" possono essere diverse: si va da una posizione dai lineamenti anarchici che non riconosce nessuna legge, e dunque non ne propone ("*nessun diritto, nessun dovere*"), a critiche più caute basate sulla ovvia constatazione che molte delle campagne *liberal* statunitensi, come quella sul *blue ribbon* o contro il *clipper chip* (standard di crittografia "debole" proposto dal governo USA in alternativa alla crittografia "forte" di programmi come il PGP), vengono di fatto sostenute dalle grandi imprese informatiche il cui principale obiettivo è quello di tutelare le proprie possibilità di commercio, più che la libertà di espressione in sé stessa.

Questo libro vorrebbe aiutare anche a osservare da una diversa prospettiva questo gran calderone di libera espressione, reti, censure, leggi, lobbies e militanza politica. È una prospettiva che nasce dall'incontro di due "linee di fuga": da una parte una certa esperienza di "vita in rete" e una certa competenza tecnica, ingredienti che permettono una familiarità con il ciberspazio nei suoi diversi aspetti simbolici e antropologici, così come informatici e relativi alla (in)sicurezza dei sistemi telematici. Si tratta com'è ovvio di esperienze e competenze vissute e guadagnate in prima persona, spesso negli anfratti più bui e nascosti della rete, che quindi consentono di porsi a una certa divertita distanza dalle rappresentazioni di Internet o delle BBS che vengono fatte al grande pubblico. In secondo luogo, gli autori di queste pagine condividono (pur nelle differenze che li separano altrove) una sorta di atteggiamento *hands on* - un atteggiamento hacker, alla "*mettiamoci le mani sopra*" - che può prendere forma ad esempio nello scrivere in proprio i programmi di cui si ha bisogno o comunque nel rendersi conto che una cosa simile, con un po' di determinazione e pazienza, è alla portata di chiunque.

Questo atteggiamento hacker può essere ovviamente applicato anche alla sfera politica delle libertà personali. Nessun riconoscimento delle autorità, nessuna delega per quanto riguarda le decisioni inerenti la propria esistenza, nessuna fiducia nei provvedimenti legislativi di tutela dei "diritti" e nella giustizia che li dovrebbe applicare. Al contrario, una insoddisfabile curiosità, una forte disponibilità ad assumere le responsabilità in prima persona, una

spinta a trovare soluzioni creative a quelli che vengono avvertiti come bisogni da soddisfare, prima ancora che come "diritti" da reclamare.

Il risultato è una posizione che non esclude *necessariamente* la militanza e l'attivismo politico tradizionale, e che non nasconde le possibili differenze tra una legge e un'altra - ma che nondimeno si pone su un piano totalmente e irriducibilmente diverso da quello del dialogo istituzionale.

Cypherpunk & Cryptoanarchy

> *L'unica conseguenza di qualsiasi legge sulla privacy è di rendere più piccole e più invisibili le microspie e le altre tecnologie di controllo.* (Robert Heinlein)

Alcuni negozi specializzati negli Stati Uniti cominciano già a vendere, a prezzi abbordabili, telecamere per il controllo a distanza non più grandi di mezzo pacchetto di sigarette. Le telecamere a circuito chiuso piazzate in punti strategici delle grandi città sono sempre più diffuse, in Inghilterra ad esempio sono già attivi diversi progetti di monitoraggio urbano su vasta scala. E se questo è quello che accade nel mondo fisico, in rete le potenzialità di controllo stanno seguendo le stesse direzioni.

Di fronte a tutto questo, abbiamo detto, ci si può indignare, si possono indire manifestazioni di protesta, si possono proporre nuove leggi a tutela della privacy. Ma non si può dimenticare il fatto che la tecnologia è come l'informazione: *non è reversibile.* Non si può "tornare indietro", non si può "dimenticare" l'informazione o la tecnologia. L'irreversibilità della scienza, della tecnologia e dell'informazione è una cosa di cui l'uomo si è accorto pienamente a partire dallo sgancio della prima bomba atomica su Hiroshima: è da quel momento che il genere umano si è reso conto per la prima volta di possedere la capacità di distruggere il pianeta, di non poter recedere da questa possibilità e quindi di dover imparare a convivere con essa. Questa convivenza può basarsi di volta in volta sulla paura (come nella corsa agli armamenti), sulla sopraffazione (come nell'odierno "nuovo ordine mondiale"), sul calcolo, su accordi internazionali o su qualche tipo di inibizione morale - certamente non sulla legge: nessuna legge ha potuto proibire agli americani di sganciare la bomba su Hiroshima e nessuna legge ha il merito di aver finora impedito le guerre nucleari.

Allo stesso modo, qualunque legge che intenda regolamentare l'utilizzo delle tecnologie di controllo avrebbe come unico risultato quello di circoscrivere l'accesso a queste tecnologie a settori privilegiati della società: detto in termini poco eleganti, a chi possiede i soldi o il potere per permettersele, ai ricchi e alle agenzie di controllo istituzionali (polizia, militari e servizi segreti); la legge italiana 675 del 31 dicembre 1996 sulla "tutela" dei dati personali sembra avere ampiamente confermato questo principio, prevendo alcune categorie di persone "particolari" (come giornalisti e, ovviamente, poliziotti) cui vengono riservate speciali deroghe rispetto a quanto è permesso ai comuni cittadini.

Anche ragionando nella migliore delle ipotesi, una improbabile legge "ideale", sostenuta da vasti movimenti di opinione ed approvata da un parlamento "illuminato", potrebbe ottenere, come massimo risultato, quello di limitare l'accesso alle tecnologie di controllo alla sola polizia. Ma perfino in questo caso una simile prospettiva potrebbe rallegrare solo chi non si è ancora accorto di come qualunque polizia del mondo abbia sempre *sistematicamente* e sotto ogni punto di vista *abusato* dei propri poteri.

Per questo motivo riteniamo che la distinzione tra leggi "buone" e leggi "cattive" vada inserita su un piano di discussione differente e, per chi vuole, parallelo rispetto a quello portato avanti qui. Che è quello di una soddisfazione *unilaterale* dei propri bisogni di privacy e di libertà individuali, che non passa attraverso i meccanismi della rappresentanza democratica, dei partiti, delle leggi, dei giudici e dei poliziotti. Abbiamo parlato di "bisogni" di privacy e di libertà, non di *diritti*, perché troppo spesso ci si riduce a vedersi elargiti i propri "diritti" da qualche magnanimo sovrano (più o meno democratico a seconda dei casi). Con questo libro proponiamo invece una serie di strumenti con cui privacy e libertà personali, limitatamente al ciberspazio (ma è ovvio che ci piacerebbe veder esteso questo principio anche altrove), diventano *appropriazioni individuali unilaterali*.

Tutto questo ci conduce inevitabilmente a chiarire il nostro rapporto con la tecnologia. Ottimismo e pessimismo a questo proposito ci sembrano ugualmente distanti: non crediamo al potenziale "liberatorio" della tecnologia così come non crediamo che la tecnologia sia necessariamente strumento di dominio: libertà e dominio sono categorie che riguardano gli uomini e non le macchine.

Vorremmo insomma uscire dalle opposte versioni del determinismo tecnologico: da una parte gli entusiasti nuovi ricchi (come gli adepti della - peraltro spesso interessante - rivista americana *Wired*, la nuova "classe virtuale") che pensano che la tecnologia ci renderà tutti più liberi e felici; dall'al-

tra parte i "naturisti" timorati di dio, che evitano la tecnologia per motivi essenzialmente ideologici: perché è, appunto, strumento di dominio, di controllo, di sfruttamento, perché snaturalizza l'uomo e via di questo passo. Per quanto ci riguarda, rifiutiamo la distinzione tra condizione "naturale" e cultura/tecnologia. Rifiutiamo le ideologie e intendiamo appropriarci della conoscenza di qualsiasi cosa ci sembri utile o semplicemente divertente - e in questo caso siamo convinti ci sia in gioco molto di più dell'utilità e del divertimento.

Certo, le storie personali di alcuni di noi ci portano talvolta a sentirci più vicini a chi diffida della tecnologia sottolineando la sua funzionalità al dio della produzione. Pur rimanendo indifferenti all'invocazione di un "ritorno alla natura", potremmo spingerci allora fino a proporre una lettura di questo libro che sia addirittura compatibile con la pratica e il pensiero *luddista*. Tralasciando le descrizioni, patetiche e ingenue, che dei luddisti sono state fatte ad opera della storiografia ufficiale, apprendiamo che i luddisti - bande di uomini mascherati e anonimi - non erano contrari alla tecnologia in sé stessa, quanto piuttosto ai mutamenti sociali che la nuova tecnologia rifletteva.

"*Le fonti popolari raccontano di Capitan Swing* [sorta di luogotenente del Generale Ludd, quest'ultimo "supereroe" mitico ante-litteram costruito dai luddisti sulla base di un personaggio, pare, realmente esistito alcuni decenni prima n.d.t.] *e banda vestiti da gentlemen che viaggiano per le campagne su calessi verdi, fanno misteriose domande sulla misura dei salari e sulle trebbiatrici, distribuiscono denaro e danno fuoco ai pagliai con pallottole incendiarie, razzi, palle di fuoco e altri congegni diabolici.*" (Luigi Bontempi, *Generale Ludd e Capitan Swing*, Nautilus, 1996, p. 28)

Congegni diabolici. È un peccato che gli attuali eredi di Ludd si siano fermati qui, cogliendo raramente l'occasione per esplorare più a fondo questo concetto e applicarlo alla società digitale. La nostra speranza è che i prossimi capitoli possano offrire anche a loro qualche idea utile...

Dalla teoria alla pratica

Superata questa introduzione ad ampio raggio sulla vita in rete, i prossimi capitoli si addentreranno negli aspetti più specifici e tecnici di alcune *armi non convenzionali* per l'autodifesa della sfera individuale nella società digitale. Da ciò che si è detto finora, però, dovrebbero a questo punto emergere anche alcuni dei risvolti immediatamente politici di queste tecniche. Alcuni li abbia-

mo accennati nelle pagine precedenti parlando ad esempio di crittografia, altri li lasciamo all'immaginazione (o alle necessità) di chi legge. In ogni caso vorremmo sottolineare in particolare due caratteristiche di queste "armi", che costituiscono un po' anche il senso unificante di questo lavoro:

1. sono utilizzabili individualmente e unilateralmente - non richiedono la mediazione di partiti o associazioni

2. il loro uso difende la sfera individuale e si affida alla responsabilità del singolo - detto diversamente, il loro uso può risultare "sociale" o "antisociale" a seconda delle circostanze, dei punti di vista e degli utilizzatori stessi.

La crittografia a chiave pubblica costituisce uno strumento fenomenale, che costituisce spesso anche l'elemento di base su cui vengono costruiti marchingegni più sofisticati come gli anonymous remailer e i nym server. Per questo motivo le verrà dedicata particolare attenzione: il prossimo capitolo provvederà a chiarire i concetti elementari della crittografia a chiave pubblica e presenterà il famoso PGP, il software diventato ormai uno standard mondiale in questo campo.

Seguirà una pratica guida all'installazione e all'utilizzo di questo software, completa dei comandi di base e di alcuni suggerimenti per risolvere i problemi più comuni che si presentano solitamente agli utilizzatori inesperti.

Il capitolo sui *file system crittati* illustrerà alcune semplici ma efficaci applicazioni di crittografia e steganografia su intere porzioni del nostro disco fisso. Avete mai pensato, ad esempio, di "minare" il vostro computer? Se siete in qualche modo soggetti a rischio di spionaggi, perquisizioni o sequestri, potrebbe essere una buona idea.

Seguirà una parte relativa agli *anonymous remailer* e al loro corretto utilizzo, che è meno banale di quanto si pensi a prima vista. Infatti, affinchè l'anonimato non sia affidato solo alla "buona fede" di chi gestisce il remailer ma sia invece verificabile oggettivamente e indipendentemente (o quasi) da fattori umani, è necessario che i remailer vengano utilizzati in catena e seguendo procedure molto rigorose.

Il capitolo sui *nym server* presenterà questi sistemi, evoluzione dei remailer tradizionali, il cui scopo è quello di offrire la possibilità di una comunicazione anonima ma anche *bidirezionale*. Si tratta di una caratteristica anti-intuitiva, ma relativamente semplice da afferrare una volta che si siano compresi i meccanismi di base della crittografia a chiave pubblica e dei remailer concatenati.

Proseguiremo con un capitolo dedicato alla *steganografia*; se con il PGP impediamo al nemico di capire cosa stiamo dicendo, con la steganografia - l'arte della scrittura nascosta - possiamo addirittura trasformare i nostri crudeli piani di conquista del mondo in innocentissime immagini, suoni, o perfino discorsi di pace e di fratellanza. Si tratta di una tecnica interessantissima, ancora poco conosciuta e in rapido sviluppo, che mette totalmente e *irrimediabilmente* fuori gioco lo Stato da qualsiasi possibile regolamentazione, presente o futura, della crittografia. Un'intuizione geniale, che una volta compresa e utilizzata ci farà accogliere con sorrisini sarcastici ogni nuova proposta di legge, come quelle recentemente giunte dall'Inghilterra o dalla Comunità Europea, di messa al bando della crittografia "robusta".

Telefonia digitale crittata: come fottere la Telekom e lo Stato in un colpo solo - e per di più (per chi ci tiene) in modo del tutto legale. Ovvero, come parlare (a voce) in privato con gli amici, attraverso computer, modem, scheda audio e software di crittografia in tempo reale - e se non basta, come farlo in tariffa urbana via Internet.

Per concludere, ci liberiamo anche dei cavi e ci spostiamo nell'etere attraverso il *packet radio* e le comunicazioni wireless: d'ora in avanti gli sbirri annuseranno un po' l'aria e sapranno che potremo essere anche nelle loro case...

Bibliografia ragionata

Questa Introduzione riassume alcuni degli eventi più significativi accaduti in rete negli ultimi anni. È ovvio che si tratta di un punto di vista necessariamente parziale. Chi fosse interessato ad approfondire questi o altri eventi, può affidarsi a una letteratura ormai pressochè sterminata: l'argomento Internet e reti telematiche hanno infatti prodotto in pochissimi anni una quantità di libri sconcertante. Nella maggior parte dei casi, questi libri costituiscono semplicemente operazioni editoriali costruite per cavalcare la moda e ritagliarsi un pezzetto di notorietà senza avere nulla di particolare da dire.

Tralasciando quindi gli instant-book "usa e getta", rimangono comunque diversi lavori interessanti che vale la pena consultare. Non dovrebbe esserci bisogno di dire che quelli elencati di seguito sono solo dei suggerimenti da cui partire per costruirsi un proprio percorso di approfondimento.

Sulla storia di Internet e dei protocolli di comunicazione non ci sono opere "definitive", ovviamente perché tale storia è tutt'altro che conclusa. Notizie

utili e documentate sugli aspetti storico-sociali delle reti si trovano in "*Internet, Pinocchio e il gendarme*" di Franco Carlini, Manifestolibri, 1996. A proposito di reti telematiche amatoriali specificamente italiane e dei loro sviluppi underground, si può consultare (sorvolando sul tono buonista-pacifista che evidentemente non ci appartiene) "*Telematica per la pace*" di Carlo Gubitosa, Enrico Marcandalli e Alessandro Marescotti, Apogeo, 1995 - ma soprattutto "*Spaghetti Hacker*" di Stefano Chiccarelli e Andrea Monti, Apogeo, 1997, che con taglio quasi etnografico racchiude anche numerose testimonianze di prima mano sulla scena telematica italiana dagli anni ottanta a oggi.

Un'eccellente storia della cultura e della filosofia *hacker* d'oltreoceano a partire dagli anni cinquanta è costituita da "*Hackers: eroi della rivoluzione informatica*", di Steven Levy, Shake, 1996, monumentale raccolta di fatti e aneddoti con l'unico neo di essere poco aggiornata sugli sviluppi dell'ultimo decennio. Sempre sulla scena underground americana, ma concentrato sulla fine degli anni ottanta/inizio anni novanta e in particolare sulla vicenda Sun Devil è "*Giro di vite contro gli hacker*" di Bruce Sterling, Shake, 1993.

Sul mondo della crittografia esistono svariati trattati tecnici, molti dei quali non tradotti in italiano. Il più citato (e uno dei più accessibili a una lettura da parte di profani) è "*Applied Cryptography*" di Bruce Schneier, Wiley & Sons, 1994. Chi vuole limitarsi a una trattazione del connubio tra crittografia (e tecnologia in generale) e strategia militare può rivolgersi a "*La guerra nell'era delle macchine intelligenti*" di Manuel De Landa, Feltrinelli, 1996.

Sulle campagne per i diritti civili e sull'organizzazione pratica di forme di attivismo telematico, una fonte variegata è "*Net Strike, NoCopyright, Et(-:*" a cura di Strano Network, AAA, 1996. Sui problemi generali della sorveglianza e del controllo nella società digitale, "*L'occhio elettronico*" di David Lyon, Feltrinelli, 1996.

Infine, sul luddismo vi sono naturalmente numerosi testi storici. Alcune brevi considerazioni su Ludd e il mondo telematico si trovano in "*Generale Ludd e capitan Swing*" di Luigi Bontempi, Nautilus, 1996.

Oltre a questi libri "su carta" esiste un'importantissima fonte di informazione costituita dalla rete stessa. Internet e le BBS pullulano di newsletter, manoscritti elettronici, forum di discussione. Anche in questi casi, né più né meno che per i libri stampati, si tratta per la maggior parte di chiacchiere inutili che rendono più ardua la scoperta delle fonti realmente interessanti. Fonti di informazione valide e affidabili sono comunque i vari newsgroups "storici" sulla crittografia (tra questi sci.crypt e alt.security.pgp) e il Computer underground Digest, un settimanale elettronico che da anni riporta con costanza le cronache più importanti dal mondo delle reti.

Ci si renderà conto che in queste pagine e in quelle che seguiranno abbiamo limitato al massimo i riferimenti a specifici indirizzi di rete e a specifiche pagine web, per non appesantire la lettura con indirizzi incomprensibili e di scarsa importanza, dal momento che le risorse in rete sono costantemente in movimento e quasi tutti gli indirizzi cambiano di solito in tempi molto brevi. Per quanto possibile, abbiamo cercato di concentrare l'attenzione soprattutto sulle questioni generali, piuttosto che su singoli software o pagine web. In compenso, questo libro ha una sua appendice telematica in rete, che costituisce un po' la sezione "dinamica" di questo lavoro. Le pagine web di "Kriptonite" riportano gli indirizzi di rete aggiornati di tutte le risorse citate nel libro, oltre a notizie sugli autori, chiavi pubbliche e altre informazioni. Rimandiamo quindi al seguente URL:

```
http://www.ecn.org/kriptonite
```

Infine, una delle prime cose che impara qualunque nuovo navigatore è l'utilizzo dei motori di ricerca, che restituiscono indirizzi aggiornati sulla base delle parole chiave immesse dall'utente: a questi *search engines* rimandiamo per la localizzazione in rete di ulteriori risorse.

Beh, amico. A questo punto le palle mi cominciavano a ruotare. Eccheccazzo, pensavo, il vecchio Luthor si è rincoglionito?

Usare Internet per tenerci in contatto. Questo un suo senso ce l'aveva. Ma perché mi faceva star lì a leggere le stronzate di 'sti pazzi furiosi? "Professionisti del controllo sociale" "Determinismo tecnologico" "Esplorare più a fondo questo concetto e applicarlo alla società digitale". Bei paroloni, sì. Da slogarmici la bocca. E intanto mentre me ne sto qua seduto con la bocca slogata l'FBI, la CIA, la pula, l'Esercito e pure gli accalappiacani hanno il solo scopo nella vita di trovare me, e vanno in giro ad annusare il buco del culo di mezzo mondo. OK, quelli non è che mi facevano troppa paura, senza il libretto delle istruzioni non sanno nemmeno sbottonarsi la patta per pisciare, te lo dico io. Ma Superman è tutto un altro paio di palle. Quello si era già levato in volo, credimi. E stava cercando la Bomba a tutt'andare. Perlustrazione a spirale ad ampio raggio con tutti i SuperSensi attivati, peggio di un mastino che fiuta una cagna in calore. Ma per fortuna alle fogne non ci si è nemmeno avvicinato. Forse nemmeno ci ha pensato, e poi al nostro amichetto SuperDotato mica gli piace troppo entrare nelle fogne. Ci tiene a essere tutto pulitino e profumatino lui, quando si presenta come Clark Kent ad annusare la gonnella della sua fighetta giornalista con la puzza sotto il naso. Ah sì, hai ragione, lo sa tutta la città che lui è Clark Kent, mica solo la sua donna e quell'altro frocetto di Jimmy Olsen. Solo che nessuno glielo ha mai fatto capire di saperlo. "Per non metterlo in imbarazzo", dicono loro. E lui crede davvero che nessuno si sia mai accorto che ogni volta che Clark Kent scappa via come avesse la sciolta che gli straborda dai pantaloni, tre secondi dopo sono tutti col naso in aria a guardare SuperMan che sfreccia nel cielo come un jet della Guardia Nazionale.... Ahahahahah! sì, è proprio un SuperImbecille. Tutto muscoli e niente cervello, lo dico sempre anch'io. Ma amico, a quei muscoli comunque devi starci ATTENTO, pure questo dico sempre. E l'idea di quella mezza tonnellata di SuperMuscoli che volteggiava in cielo cercando proprio me, un certo pizzicorino alla nuca cominciava a farmelo venire. E per quanto sia un idiota, ha del cervello di scorta. La sua

Lois e il suo frocetto, mica li puoi pigliare troppo sottogamba quando attaccano ad andare in giro a fare domande. Lei è furba, conosce tutta la città, e in più c'ha due tette da rincoglionire un eunuco in pensione e sa come metterle in mostra. E anche lui, per quanto finocchietto, stupido non è. E anche senza prove, a sospettare che solo Lex Luthor poteva avere messo su un colpo del genere non ci voleva mica molto. E a indovinare che se c'entrava Lex allora c'entrava anche il qui presente, ci poteva arrivare chiunque fosse abbastanza furbo da sapersi allacciare le scarpe da solo. Gli accalappiacani stavano girando a vuoto con la loro solita solfa. Gli Arabi, i Terroristi Internazionali, gli Estremisti, la Mafia. Gente metodica, che si ripassa tutti i sospetti, prima di arrivare a me ci avrebbero messo una vita. Ma quel duo mi preoccupava. E Lex, furbone, col culo in caldo al sole delle Bahamas non correva nessun rischio. Il Bottone ce l'avevo io, mica lui. E il miliardo, se tutto andava liscio, ce l'aveva lui, mica io... Lì per lì mi era venuta la voglia di piantar tutto e mandare il vecchio Lex affanculo. Quelle scarpe di cemento sono pesanti, sì. Ma pesano troppo per rincorrermi con quelle per tutto il mondo. Anche per lui, mi capisci? Comunque, era una cosa da pensarci bene. E allora, mentre ci pensavo, tanto valeva vedere dove voleva arrivare il vecchio Lex con quella storia. Che cazzo dovevo fare? Mi sono rimesso a leggere...

Crittografia

di Luc Pac

Chiunque si interessi di questioni inerenti la privacy digitale avrà sicuramente già sentito parlare del software *Pretty Good Privacy*. Il PGP (così è familiarmente chiamato da tutti), come accennato nell'Introduzione, è un programma informatico di crittografia che si è diffuso in tutto il mondo dai primi anni '90; funziona praticamente su qualsiasi computer ed è completamente gratuito. Questo capitolo cercherà di esporre un'introduzione pratica alla crittografia e di spiegare gli aspetti più generali dell'utilizzo di un programma come il PGP, rimandando al prossimo capitolo le questioni relative all'installazione e ai comandi più comuni. È importante sottolineare che la lettura di queste pagine non sostituisce e non deve sostituire un'attenta e ripetuta lettura del manuale originale allegato al PGP stesso: qualsiasi utilizzo del PGP che prescinda da questa lettura costituisce un potenziale pericolo per sé e per le altre persone con le quali si comunica!

Algoritmi e chiavi

Crittografia: s.f. sistema segreto di scrittura in cifra o codice
(dal Vocabolario della lingua italiana di Nicola Zingarelli)

La crittografia è un'arte antichissima: consiste nel rendere incomprensibile un certo messaggio a occhi estranei. Ecco un sistema crittografico elementare:

a	b	c	d	e	f	G	h	i	j	k	l	m
1	2	3	4	5	6	7	8	9	10	11	12	13

n	o	p	q	r	s	t	U	v	w	x	y	z
14	15	16	17	18	19	20	21	22	23	24	25	26

Ad ogni lettera di questo alfabeto corrisponde un numero da 1 a 26. Basta mettersi d'accordo con il proprio interlocutore, affinché nei messaggi ogni lettera venga sostituita dalla lettera che la segue di 13 posizioni. In questo modo la lettera "a" (posizione 1) viene sostituita dalla lettera "n" (posizione 14), e così via. Utilizzando questa convenzione, il messaggio:

"il piano per la fabbricazione della kriptonite e' dietro il quadro in salotto"

diventa

"vy cvnab cre yn snooevpnmvbar qryyn xevcgbavgr r' qvrgeb vy dhnqeb va fnybggb"

In questo sistema di crittografia, la trasposizione per ogni lettera di un numero fisso di posizioni rappresenta ciò che viene chiamato *algoritmo*, mentre il numero di posizioni di cui trasporre le lettere (numero che ovviamente può cambiare di volta in volta) rappresenta la *chiave* di cifratura (o di codifica). *Algoritmo* e *chiave* sono le due componenti principali di ogni sistema di crittografia, componenti che permettono il passaggio dal *messaggio in chiaro* al *messaggio cifrato* (o *crittato*) e viceversa.

L'esempio sopra riportato costituisce un sistema di crittografia estremamente debole: un eventuale estraneo che volesse intercettare e comprendere la comunicazione (colui che viene convenzionalmente indicato con il termine di *nemico*) raggiungerebbe il suo scopo sfruttando gli evidenti punti deboli dell'algoritmo. È da notare, ad esempio, che il messaggio cifrato conserva molti indizi del messaggio in chiaro: ha lo stesso numero di lettere, mantiene le stesse separazioni tra le parole, conserva la stessa distribuzione statistica delle varie lettere. Un nemico sufficientemente scaltro impiegherebbe pochi secondi a sospettare che l'algoritmo utilizzato sia proprio una semplice trasposizione di lettere; una volta individuato l'algoritmo, diventa molto semplice e veloce anche provare tutte le 25 chiavi possibili fino a quando non appare un messaggio comprensibile.

La crittografia, nel corso dei secoli e data la sua importanza in ambito militare e strategico, è diventata quindi anche una tecnica estremamente complessa capace di utilizzare algoritmi derivati dalle più avanzate conoscenze nel campo della matematica. Con la progressiva crescita di complessità degli

algoritmi, si è arrivati anche a definire alcuni requisiti di base che qualsiasi sistema crittografico deve soddisfare affinché possa essere considerato sufficientemente *robusto* (cioè difficilmente attaccabile da tentativi di *crittanalisi* da parte del nemico): uno di questi requisiti è che la robustezza del sistema non deve dipendere dalla segretezza dell'algoritmo (il cosiddetto "principio di Kerckhoff"). Nel nostro esempio, una volta che il nemico individua l'algoritmo di trasposizione delle lettere gli è sufficiente provare al massimo 25 combinazioni (chiavi) diverse per avere la certezza di risalire al messaggio in chiaro. Nei sistemi crittografici più robusti, invece, anche la totale conoscenza dell'algoritmo non permette in nessun modo di comprometterne la sicurezza, unicamente affidata alla segretezza della chiave concordata (che ovviamente avrà un campo di variazione potenzialmente infinito).

Per questo motivo, nella delicatissima fase di valutazione dell'affidabilità di un sistema crittografico, vengono presi in considerazione da parte della comunità *criptoanarchica* internazionale soltanto quei sistemi di cui gli autori hanno messo a disposizione tutti gli algoritmi impiegati; nel caso di software informatico, gli algoritmi sono rappresentati dai codici sorgenti, che permettono a chiunque di decostruire e ricostruire il software sul proprio computer.

Una prima considerazione da fare a questo punto è la seguente: la maggior parte dei sistemi di crittografia proposti o utilizzati dagli enti governativi sono basati su algoritmi mantenuti segreti. Il PGP, al contrario, utilizza algoritmi pubblici e ampiamente conosciuti e studiati, e nonostante questa dimensione pubblica garantisce un'ottima sicurezza. Già solo questo punto spinge a diffidare di qualunque sistema crittografico proposto dallo Stato e a considerare allo stesso tempo il PGP come un sistema privo di difetti evidenti (che sarebbero immediatamente stati scoperti dalle migliaia di ricercatori che hanno analizzato il PGP nel corso di questi anni, ognuno dei quali avrebbe guadagnato fama pubblica, ricchezza economica e carriera accademica da un eventuale successo in questo senso).

Il motivo per cui viene consigliato proprio il PGP, tra i tanti programmi disponibili, come software di crittografia "preferito" è semplice: si tratta di un programma sviluppato praticamente in pubblico, alla luce del sole, sotto gli sguardi attenti di una vasta comunità di esperti; inoltre viene distribuito completo di sorgenti, che permettono - a chiunque ne abbia voglia o capacità - di cimentarsi con lo studio dei suoi algoritmi. Il PGP pare dunque essere un programma che non ha paura di mostrare in pubblico quali potrebbero essere i suoi punti deboli; mentre al contrario, sui sistemi crittografici "proprietari" fioriscono continuamente leggende e illazioni (fondate o meno) su possibili backdoors - "porte sul retro" - che garantirebbero solo a chi le conosce una facile decodifica dei messaggi.

È anche il caso di accennare a una cosa che dovrebbe essere scontata, ma che è comunque bene chiarire una volta per tutte: la crittografia impiegata dal PGP e in generale il livello di sicurezza offerto dalle varie tecniche di cui si parlerà in questo libro non ha nulla in comune con le opzioni di "sicurezza" offerte da vari software per ufficio usati comunemente. Molti di questi programmi (word processors, databases, archiviatori e compattatori, eccetera) offrono infatti la possibilità di proteggere i file con una password; si tratta quasi sempre di protezioni debolissime che possono essere superate in pochi minuti da persone sufficientemente esperte. I livelli di privacy e sicurezza che ci interessano non sono quelli di questi giocattoli e non sono nemmeno quelli degli standard di crittografia previsti in genere dai governi nazionali per l'utilizzo da parte dei privati cittadini: questi standard possono offrire una discreta protezione da eventuali attacchi condotti da singoli privati, ma sono comunque superabili in caso di reale necessità da parte delle agenzie di sicurezza governative. Le tecniche presentate qui sono da considerarsi alla stregua di armi da *guerriglia digitale*, in grado di resistere ad attacchi militari seri e impegnati.

Sulla sicurezza del PGP si tornerà più avanti. Ora è il momento di parlare di un'altra particolare caratteristica di questo programma. Com'è noto, il PGP è un software di crittografia *a chiave pubblica*.

Crittografia a chiave pubblica

Joe Lametta vorrebbe mandare una bottiglia di vino rosso italiano a Lex Luthor, ma sa che un raffinato postino potrebbe intercettare il pacco, aprirlo e bere il contenuto della bottiglia. Luthor è in vacanza alle Bahamas - e non può andare personalmente a ritirare il pacco da Joe Lametta. Come può fare Joe per inviare il pacco a Lex - nella sicurezza che nessun altro tranne il destinatario possa aprirlo?

Esiste almeno una soluzione logica a questo problema del tutto teorico: Joe chiude il pacco con un lucchetto di cui possiede la chiave. Conserva la chiave e spedisce il pacco a Lex. Quest'ultimo riceve il pacco (che non può aprire perché non ha la chiave). Lo chiude nuovamente con un lucchetto di cui conserva a sua volta la chiave e rispedisce il pacco a Joe. Joe toglie il suo lucchetto e rispedisce il pacco a Lex - che può finalmente aprire la sua bottiglia di vino, chiusa ormai solo con il lucchetto da lui scelto.

Questo semplice metodo condivide alcune caratteristiche con la crittografia a chiave pubblica: si tratta di un sistema che risolve efficacemente il classico problema della crittografia tradizionale. Se la sicurezza del sistema dipende dalla segretezza della chiave di codifica utilizzata, allora è necessario almeno un canale sicuro attraverso il quale trasmettere la chiave. Per concordare una chiave con il proprio interlocutore c'è bisogno di mettersi preventivamente in contatto con lui incontrandolo di persona, telefonandogli, scrivendogli una lettera, mandandogli un messaggero o in qualsiasi altro modo. In qualsiasi caso, esiste il pericolo che la chiave venga intercettata durante il tragitto, compromettendo quindi l'intero sistema comunicativo.

La crittografia a chiave pubblica permette a due (o più) persone di comunicare in tutta riservatezza anche se non si sono mai incontrate prima e dunque non è mai stata concordata in precedenza alcuna chiave di codifica.

La cosa è apparentemente assurda e anti-intuitiva, ma il principio in base al quale è possibile tutto questo è relativamente semplice da comprendere. Nella crittografia tradizionale viene utilizzata un'unica chiave sia per codificare, sia per decodificare i messaggi. Le informazioni (la chiave e l'algoritmo) necessarie per chi deve *inviare* il messaggio sono quindi identiche a quelle necessarie a chi deve *riceverlo*.

La crittografia a chiave pubblica, al contrario, si basa su una coppia di chiavi: una *chiave pubblica* e una *chiave segreta*. La chiave pubblica serve unicamente per *codificare* il messaggio, mentre quella segreta serve unicamente per *decodificarlo*. È come se una cassaforte avesse due chiavi distinte, una usata per aprirla e una per chiuderla. A questo punto il gioco è fatto: ogni utilizzatore del PGP si crea la propria (o le proprie, in casi particolari) coppia di chiavi. La chiave segreta viene tenuta, appunto, segreta e non viene mai rivelata a nessuno (nemmeno alle persone con le quali si comunica); viceversa, la chiave pubblica viene diffusa ovunque e in ogni modo: può essere aggiunta automaticamente in coda a ciascun proprio messaggio nelle varie conferenze elettroniche cui si partecipa, o può essere depositata in archivi pubblici (keyserver) a disposizione di chi la desideri. È importante che la tua chiave pubblica sia liberamente accessibile, perché chiunque voglia comunicare con te dovrà preventivamente munirsi della tua chiave pubblica (eventualmente anche chiedendotela direttamente) con la quale crittare il messaggio a te indirizzato.

La differenza rispetto alla crittografia tradizionale sta nel fatto che adesso non è più necessario trovare un luogo sicuro nel quale vedersi e scambiarsi la chiave di codifica/decodifica, in quanto anche nel caso la chiave pubblica venisse intercettata non ci sarebbe assolutamente nessuna conseguenza: le chiavi pubbliche che viaggiano liberamente in rete possono solo *crittare*, non

decrittare i messaggi. Con le coppie di chiavi della crittografia a chiave pubblica diventa possibile condurre discorsi riservati anche tra individui che non si sono mai conosciuti o visti di persona, magari perché separati da migliaia di chilometri di distanza.

Chiunque può usare la chiave pubblica del destinatario per crittare un messaggio diretto a lui - mentre il destinatario userà la propria corrispondente chiave segreta per decrittare quel messaggio. Nessuno che non sia il destinatario può decrittarlo, perché nessun altro ha accesso alla chiave segreta; nemmeno la persona che ha mandato il messaggio potrà più aprirlo una volta crittato.

Ovviamente questo principio generale ha una solida base matematica che lo giustifica; tale base, riassunta e semplificata all'estremo, si fonda sull'inesistenza – nel corpo di conoscenze sulla matematica pura sviluppate dall'uomo negli ultimi secoli – di un metodo sicuro e rapido per fattorizzare (ridurre ai numeri primi che lo producono se moltiplicati tra loro) un numero qualsiasi composto da un numero di cifre sufficientemente alto. In altre parole, se si prendono due numeri primi a caso X e Y, si può agevolmente moltiplicarli tra loro, ottenendo Z; ma è estremamente difficile compiere il percorso inverso: dato Z, non esiste un metodo efficace per risalire a X e Y. Se questi numeri sono scelti in modo da essere sufficientemente alti (composti cioè da centinaia di cifre), la difficoltà di risalire a X e Y non può essere superata con le attuali capacità di calcolo né con quelle ragionevolmente prevedibili nel prossimo futuro. Se è chiaro che non è questa l'occasione per addentrarsi nei dettagli di questo principio, è però necessario fare presenti due cose.

Prima di tutto, l'intera struttura matematica su cui si regge la crittografia a chiave pubblica (e quindi anche il PGP) è completamente a disposizione di chiunque abbia voglia, tempo e competenze per verificarla e metterla alla prova. Comprendere e verificare il principio di base è una cosa alla portata di chiunque abbia una buona base matematica a livello scolastico. Addentrarsi negli algoritmi veri e propri, certamente, richiede qualcosa di più. In ogni caso, l'intero corpo di conoscenze necessario è di dominio pubblico e facilmente recuperabile in qualsiasi biblioteca ben fornita. Se qualcuno ha dei dubbi sull'efficacia del PGP e non si fida degli anni e delle migliaia di ore spese ad analizzarlo pubblicamente da parte di persone interessate e motivate di tutto il mondo, alcune delle quali tutt'altro che "comode" per l'establishment e con motivi tutt'altro che deboli per cercare un sistema di crittografia veramente robusto, ebbene questo qualcuno non ha che da rimboccarsi le maniche e fare un po' di verifiche in prima persona: troverà tutte le porte spalancate.

In secondo luogo, è fondamentale ricordare che il PGP *non è un sistema assolutamente sicuro*. Nessun sistema di crittografia lo è, né lo potrà mai essere.

La sicurezza che può offrire un ottimo sistema di crittografia come il PGP è solamente *relativa*, per quanto molto vicina a quella assoluta: essa dipende da una serie di fattori (tecnologici, matematici, umani), di cui però l'elemento di gran lunga più importante rimane l'attenzione dell'utilizzatore umano. È perfettamente inutile speculare sui milioni di anni di calcolo necessari a spezzare una chiave RSA, se poi la propria chiave segreta e la propria frase di accesso rimangono a disposizione di chiunque sulla scrivania o sulla propria agenda.

Sulla questione degli "attacchi pratici", indipendenti dalla robustezza degli algoritmi di crittografia usati, si tornerà tra poco in quanto si tratta di un argomento importantissimo e spesso trascurato, ma nel quale la creatività e la fantasia degli esseri umani talvolta riescono a superare anche le più ardue barriere matematiche.

Firme digitali e pseudonimato

Facciamo ora una breve digressione e analizziamo una delle caratteristiche più evidenti della comunicazione in rete in generale.

Le interazioni in rete avvengono necessariamente fra "entità" e la dimensione più strettamente fisica viene parzialmente messa da parte. Questo però non significa affatto, come si sente dire da più parti, che le interazioni siano condotte per questo in forma necessariamente *anonima*.

L'oggettiva difficoltà di risalire con certezza al collegamento biunivoco tra "entità" interagente e individuo fisico che muove quell'entità costituisce sempre più un problema da un punto di vista giuridico: nonostante in rete possano essere compiuti reati, truffe e illeciti di vario tipo, la responsabilità legale di questi atti ricade spesso nel nulla, in quanto è molto difficile accertare quale persona fisica stia effettivamente operando all'altro capo del filo in quel preciso istante. Ma questi sono problemi giuridici e in particolare della giurisdizione statale (la particolare "giurisdizione" sviluppata spontaneamente in rete pare non preoccuparsene troppo). È solo in un'ottica repressiva e di controllo che, partendo da questa considerazione, si può arrivare a concludere che la comunicazione in rete è sempre anonima. Ciò che manca in rete non è il nome delle persone, ma semplicemente la zavorra dell'*identità anagrafica*.

A sostituzione del nome anagrafico, assegnato per legge e immodificabile, in rete prolifera una quantità enorme di altri nomi, *nicks*, *handles*, *alias*, pseudo-

nimi. L'importanza di questi nomi non è minore di quella del proprio nome anagrafico: è attraverso il loro riconoscimento pubblico che in rete è possibile costruire relazioni sociali significative, che potranno ovviamente poi essere trasferite anche al di fuori della rete.

Se esiste dunque una modalità caratteristica dell'interazione in rete rispetto ai nomi e alle identità individuali, questa non è data principalmente dall'anonimato ma piuttosto dallo *pseudonimato*. Lo pseudonimato comporta un processo di costruzione dell'identità e un suo riconoscimento sociale che perdurano nel tempo ma sono anche mutevoli e continuamente in divenire, mai acquisiti definitivamente; patrimonio fondamentale dello pseudonimo (sia esso corrispondente o no a un nome anagrafico) è la *reputazione* che esso riesce a guadagnare attraverso la sua vita in rete o quella che eredita da un'eventuale ragnatela di relazioni sociali avviate in precedenza off-line.

La perfetta realizzazione dello pseudonimato si scontra però con gli stessi problemi a cui si è accennato a proposito dell'identità anagrafica: via rete è possibile modificare non solo il nome-numero di serie che lo Stato ci ha affibbiato, ma anche lo stesso pseudonimo che ci siamo scelti. È possibile scrivere firmandosi con uno pseudonimo altrui. Questa possibilità costituisce il più delle volte una sanissima e utile opera di decostruzione dei propri pregiudizi e delle proprie rigidità ed è una possibilità che ci guardiamo bene dal voler eliminare. Nonostante questo, ci sono casi in cui, magari a causa della natura molto "specifica e concreta" della comunicazione, è assolutamente necessario essere certi dell'autore di un dato messaggio. Non tanto essere certi del suo numero di serie statale, quanto piuttosto del fatto che egli è effettivamente la stessa entità (individuale o collettiva) con cui si è comunicato in precedenza, via rete o anche in carne ed ossa. In altre parole, è necessario essere certi del suo pseudonimo.

È in una situazione come questa che la crittografia a chiave pubblica viene nuovamente in aiuto. Semplicemente rovesciando l'impiego delle chiavi pubbliche e private, è possibile porre una *firma digitale crittografica* sui messaggi che immettiamo in rete. La chiave segreta del mittente può infatti essere usata, se si vuole (oltreché per decrittare i messaggi ricevuti) anche per generare una firma da apporre nel corpo dei messaggi che si spediscono. La firma digitale del messaggio può poi essere verificata dal destinatario (o da chiunque altro) utilizzando la chiave pubblica del mittente.

Questo serve a garantire che il mittente è colui che davvero ha scritto il messaggio - e che il messaggio non è stato successivamente manipolato da nessun altro, poiché solo il mittente possiede la chiave segreta per poter firmare. È tecnicamente impossibile falsificare o modificare un messaggio

autenticato senza invalidarne la firma e lo stesso mittente non può più revocare la firma una volta apposta. Detto in altre parole, il PGP ti permette di ricevere un messaggio che solo tu sei in grado di leggere e in più (se il mittente ha scelto di firmare il messaggio) ti permette di essere sicuro del fatto che quel messaggio può essere stato scritto solo da una certa persona (o comunque solo da una persona che ha accesso a una particolare chiave segreta).

Una delle applicazioni più utili della firma digitale, a parte l'autenticazione dei messaggi veri e propri, riguarda la conferma delle chiavi pubbliche di terze persone. La crittografia a chiave pubblica infatti lascia scoperto un possibile punto debole. Nel momento in cui vuoi comunicare con Lex Luthor, ti serve la sua chiave pubblica. Il modo migliore per ottenerla è direttamente dalle sue mani. Talvolta questo non è possibile e sei costretto a fartela inviare attraverso la rete. Come abbiamo visto il sistema a chiave pubblica risolve ogni problema rispetto a un'eventuale intercettazione della chiave lungo il tragitto, ma presta il fianco alla possibilità che Superman, conoscendo la tua volontà di comunicare con Lex Luthor, si spacci per lui e ti spedisca una chiave pubblica contraffatta. Se cadi nel tranello e utilizzi quella chiave, i successivi messaggi saranno leggibili non da Lex Luthor, bensì da Superman, titolare della vera corrispondente chiave segreta. Superman potrà poi perfezionare il suo inganno rispedendo a sua volta tutti i messaggi a Lex Luthor, che in questo modo non si accorgerà nemmeno dell'esistenza di una tappa in più lungo la strada.

Questo problema (chiamato problema "dell'uomo nel mezzo") è assolutamente concreto e reale. La soluzione sta nel chiedere e ottenere che ogni nuova chiave pubblica sia firmata da qualcuno che si conosce e di cui si dispone già con certezza della rispettiva chiave pubblica. Se sei costretto a ottenere la chiave di Lex Luthor via rete, avrai cura di verificare che essa sia firmata da quello sballato di Pippo, che ha contatti quotidiani con Lex Luthor. Siccome hai incontrato di persona Pippo un anno fa, e in quell'occasione vi siete personalmente scambiati le rispettive chiavi pubbliche, tu possiedi con certezza la vera chiave pubblica di Pippo. Con questa chiave puoi verificare la firma che Pippo ha apposto sulla chiave pubblica di Lex Luthor, puoi cioè verificare che Pippo, di cui ti fidi, garantisca che la chiave pubblica che ti è appena arrivata è effettivamente la chiave di Lex Luthor. Superman, da solo, non sarebbe mai in grado di mandarti una chiave firmata da Pippo, spacciandola per la chiave di Lex Luthor.

A questo punto è evidente che la certificazione delle chiavi può diventare rapidamente molto complessa (Pippo può firmare la chiave di Lex Luthor, il quale a sua volta firma la chiave di qualcun altro, tuo prossimo collaboratore), consentendoti, da un unico punto di partenza sicuro, di estendere la tua rete di

contatti a dismisura comprendendo anche entità che non incontrerai mai di persona. Con questa caratteristica il cerchio viene chiuso e diventa veramente possibile stabilire un'infrastruttura comunicativa priva di contatti fisici che sia doppiamente sicura, sia dal punto di vista della possibilità di leggere il contenuto della comunicazione, sia da quello di poterne garantire la provenienza.

Crittanalisi

Prima di concludere questa discussione generale e passare agli aspetti più concreti del PGP, è necessario un commento critico sul grado di sicurezza effettivamente raggiungibile con strumenti di questo tipo.

Si è già detto che la sicurezza dell'algoritmo non è affatto assoluta, al contrario, lo sforzo necessario per "rompere" con la sola "forza bruta" una tipica chiave RSA (questo è il nome dello specifico algoritmo di crittografia a chiave pubblica utilizzato dal PGP nelle sue versioni più diffuse) è quantificabile, con le attuali conoscenze matematiche, in modo abbastanza preciso. A seconda dei mezzi a disposizione, questo sforzo è misurato in migliaia o milioni di anni di calcolo, ipotizzando anche l'impiego di computer molto più potenti di quelli pubblicamente conosciuti in questo momento. La sicurezza, da questo punto di vista, è quindi "relativamente assoluta", *a patto che non subentrino altri anelli, più deboli, nella catena del nostro sistema.*

Scrivere la propria passphrase (necessaria per accedere alla chiave segreta) su un foglietto post-it appiccicato al monitor può ridurre i famosi milioni di anni di calcolo a pochi secondi. A parte questo esempio banale, i potenziali anelli deboli sono sfortunatamente molti. Tuttavia si tratta sempre di debolezze esterne all'algoritmo di crittografia vero e proprio: è inutile chiedersi se i milioni di anni potranno presto ridursi a centinaia con l'aiuto di computer più potenti; è molto più utile, invece, chiedersi ad esempio in base a quali criteri è stata scelta la propria passphrase. La metodologia di attacco più efficiente conosciuta fino a questo momento, infatti, piuttosto che tentare tutte le combinazioni possibili della chiave (*brute force attack*) si affida a dizionari che, una volta comunque acquisita la chiave segreta attraverso altri mezzi (vedi sotto), limitano i tentativi per trovare la passphrase alle sole combinazioni più plausibili in base a fattori tipicamente umani.

Prima di commentare i possibili attacchi condotti tramite dizionario, però, è opportuno soffermarsi ancora un momento sugli attacchi *brute force*. Nel

Dimensioni della chiave in bits	Anni-MIPS necessari per la fattorizzazione
512	30.000
768	200.000.000
1024	300.000.000.000
2048	300.000.000.000.000.000.000

caso del PGP, questi consistono nel tentativo di ottenere i codici di accesso avendo a disposizione la sola chiave pubblica. In termini matematici si tratta di "fattorizzare" un numero estremamente alto; lo sforzo necessario per compiere questa operazione dipende direttamente e in primo luogo dalla lunghezza della chiave pubblica prescelta. Le tipiche chiavi pubbliche create con il PGP sono costituite, in genere, da un numero ben preciso di bit: 384, 512, 768, 1024, 2048. All'aumentare della lunghezza della chiave pubblica aumenta anche lo sforzo necessario per fattorizzarla. Questo aumento avviene però in forma esponenziale, e quindi una chiave da 1024 bit è incomparabilmente più sicura, da questo punto di vista, di una da 512. Per rendere più evidenti queste differenze, la tabella riportata sopra indica un tentativo di stima dello sforzo richiesto, con gli algoritmi attualmente più evoluti, per fattorizzare alcune tipiche chiavi pubbliche generate dal PGP:

L'intensità dello sforzo è espressa in "anni-MIPS", una misura comunemente usata per indicare la potenza di calcolo. Un anno-MIPS corrisponde a un anno di calcolo ininterrotto effettuato da una macchina della potenza di 1 MIPS (cioè in grado di eseguire un milione di istruzioni al secondo; è da tenere presente che i comuni personal computer attualmente in commercio esprimono in genere una potenza di calcolo superiore a 200 MIPS).

Per dare un'idea concreta di quale sia la capacità di calcolo raggiungibile attualmente, ricordiamo i due più importanti eventi che hanno sottoposto a prove pratiche la resistenza del PGP agli attacchi *brute force*.

Nel primo caso si trattava di raccogliere la sfida lanciata da Ron Rivest (uno degli ideatori del sistema RSA, il sistema di crittografia a chiave pubblica su cui si basa il PGP) nel 1977: a chi fosse riuscito a fattorizzare un numero di 129 cifre (approssimativamente corrispondente a una chiave PGP da 425 bit), Rivest prometteva un premio simbolico di 100 dollari. La sfida era nota con il nome "RSA-129". Lo stesso Rivest aveva stimato che fossero necessari 40 quadrilioni di anni per realizzare una simile impresa. Nel 1994, dopo soli 17

anni, questo obiettivo è stato raggiunto seguendo una metodologia assai interessante: i calcoli sono stati eseguiti su una gigantesca macchina virtuale, composta da migliaia di computer sparsi in tutto il mondo. Infatti, anziché concentrare lo sforzo su una sola macchina, che avrebbe portato a tempi di calcolo assolutamente improponibili (anche se forse inferiori ai 40 quadrilioni di anni stimati da Rivest), la sfida è stata ampiamente pubblicizzata in rete e, attraverso alcuni coordinatori, distribuita su migliaia di volontari, ognuno dei quali metteva a disposizione le proprie risorse di calcolo nei momenti in cui queste non venivano utilizzate per le normali attività lavorative. Attraverso questa enorme iniziativa pubblica e collettiva si è riusciti, in soli otto mesi di lavoro effettivo ed esprimendo uno sforzo complessivo pari a 5.000 anni-MIPS, a superare la sfida, fattorizzando il numero magico da 129 cifre e decrittando il messaggio di sfida scritto nel 1977 da Rivest. Questo evento rese evidenti a tutti le incredibili possibilità offerte dalla cooperazione in rete su vasta scala.

Il secondo evento degno di nota è avvenuto nel 1995 e ha rappresentato il primo attacco *brute force* riuscito direttamente contro una vera e propria chiave PGP. Si trattava della chiave da 384 bit (116 cifre) di "Blacknet", un esperimento dimostrativo di come avrebbe potuto funzionare un ipotetico mercato nero di informazioni segrete, basato su crittografia e firme digitali. La chiave è stata fattorizzata in tre mesi, con uno sforzo di 400 anni-MIPS (da notare la differenza rispetto ai 5.000 anni-MIPS dell'operazione precedente, riguardante una chiave di sole 13 cifre più lunga). Nonostante l'impegno di calcolo molto più modesto rispetto a RSA-129, in questa occasione la novità era rappresentata dalla segretezza in cui si è svolto il lavoro: contrariamente alla pubblicità e agli appelli pubblici in rete alla ricerca del maggior numero possibile di volontari, in questo caso gli autori erano solamente quattro persone che hanno lavorato in privato e hanno reso noto il loro lavoro solo a operazione conclusa. Nonostante i quattro non fossero esattamente persone qualunque (erano tutti responsabili di centri di calcolo e tra le macchine utilizzate, oltre a decine di workstations, figurava anche un MasPar, uno dei computer più potenti e costosi mai costruiti) il tentativo, perfettamente riuscito, era quello di dimostrare come un attacco del genere fosse possibile anche senza pubblicità e dunque, a maggior ragione, anche da parte di agenzie governative più o meno segrete.

Questi due eventi hanno avuto il grande merito di dare finalmente una dimostrazione pubblica e concreta dell'effettivo livello di sicurezza garantito da sistemi come il PGP. I due successi nella fattorizzazione delle chiavi non devono essere interpretati affatto come un'espressione di debolezza del sistema: al contrario, ne è uscita confermata l'impossibilità pratica di attaccare frontalmente le tipiche chiavi effettivamente utilizzate in questo momento,

composte da almeno 1024 bit. Le chiavi da 512 bit, considerate sicure fino a pochi anni fa, cominciano ora a essere considerate dei possibili obiettivi da parte di ipotetiche organizzazioni dotate di enormi risorse e ovviamente di un enorme interesse nel dedicare queste risorse, per almeno alcuni mesi, a uno specifico obiettivo.

Ma i metodi offerti dalla crittanalisi (cioè la scienza che studia i modi per superare gli algoritmi di crittografia) non si limitano ai tentativi *brute force* applicati per indovinare la chiave: spesso è possibile scoprire delle debolezze matematiche nell'algoritmo, evidenti solo dopo anni di studi. Inoltre, vi sono delle particolari condizioni nelle quali il lavoro dei crittanalisti risulta facilitato. Nonostante da questo punto di vista il PGP sia reputato sicuro, può essere interessante citare qualche esempio, accennando ai tre principali scenari di attacco ipotizzati in crittanalisi, in ordine decrescente di complessità.

Il primo scenario è quello in cui il nemico non conosce nulla del contenuto dei messaggi che ci scambiamo; il nemico assiste cioè esclusivamente al transito dei messaggi nella loro forma crittografata (*cyphertext-only attack*). È questo lo scenario più sicuro, in cui molte volte l'unica possibilità di attacco è rappresentata dagli attacchi *brute force* discussi sopra. Ma se l'algoritmo è sufficientemente robusto e la chiave sufficientemente lunga, gli attacchi *brute force* risultano, come si è visto, impossibili o eccessivamente dispendiosi in termini di tempo e risorse. È normale allora che il nemico cerchi di ottenere ulteriori informazioni.

Ci spostiamo così nel secondo scenario, in cui il nemico riesce a ottenere una o più coppie di messaggi in chiaro (*plaintext*) e delle loro corrispondenti versioni crittate (*known-plaintext attack*). La possibilità di uno scenario simile non deve soprendere: è sufficiente dimenticarsi sul proprio hard disk il messaggio luthor.asc (crittato) e luthor.txt (in chiaro). Un'eventuale perquisizione e sequestro del computer potrebbe fornire al nemico molte coppie di messaggi come questa. In campo crittanalitico si cerca sempre di sfruttare la disponibilità dei messaggi in chiaro per arrivare a carpire informazioni in grado di rompere chiave e algoritmo, decrittando così qualunque altro messaggio.

Infine, un terzo scenario ancora più critico è quello denominato *chosen-plaintext attack*, in cui il nemico è in grado – normalmente attraverso l'inganno e l'astuzia – di scegliere il contenuto in chiaro di un certo messaggio e di ottenere poi la sua versione crittata. Questo tipo di attacco è ciò che ha favorito la rottura del codice "Purple" usato dai giapponesi durante la seconda guerra mondiale: le forze americane, che già monitoravano il traffico comunicativo crittato del Giappone, inviarono una particolare notizia falsa che sapevano sarebbe stata immediatamente trasmessa, crittata, al comando generale giapponese. La possibilità di avere un particolare testo scelto dal nemico (e

non un testo a caso) e la sua corrispondente forma crittata offre ulteriori informazioni utili ai crittanalisti esperti.

È bene ribadire che il PGP pare essere assolutamente resistente perfino in caso di *chosen-plaintext attack*. Tuttavia è anche bene essere consapevoli del fatto che, in linea teorica, qualsiasi sistema crittografico corre rischi maggiori quando si passa da una condizione di *cyphertext only* a una di *known-plaintext* o addirittura di *chosen-plaintext attack*. Nel caso della crittografia a chiave pubblica, poi, ulteriori rischi provengono dalle caratteristiche della firma elettronica. Lo scenario *chosen-plaintext*, infatti, può essere raggiunto da questo versante con ancora maggiore facilità: è sufficiente che il nemico ci spedisca un messaggio apparentemente banale e innocuo, chiedendoci di restituirlo dopo avervi apposto la nostra firma elettronica generata con il PGP. La nostra firma apposta su un messaggio dotato di particolari caratteristiche scelte dal nemico, in modo del tutto simile a un *chosen-plaintext attack*, può in qualche modo facilitare le operazioni di attacco al nostro sistema.

Rischi di questo tipo su un sistema crittografico basato sul PGP sono, lo ripetiamo, del tutto teorici e speculativi. Tuttavia rimane buona norma cercare di evitare, per quanto possibile, di lasciare sul proprio computer coppie di messaggi in chiaro e crittati; allo stesso modo è necessario prestare attenzione a ciò che si firma e non firmare mai contenuti ricevuti da estranei. Conoscere la possibilità di quest'ultimo tipo di attacchi può risultare utile, forse più che per proteggere un sistema già sufficientemente robusto come il PGP, per rendersi conto di quando le nostre comunicazioni cominciano ad attirare attenzioni inusuali.

Attacchi pratici

Questo per quanto riguarda le possibilità di attacco offerte dalla crittanalisi pura. Soprattutto quando i possibili nemici sono gli organi repressivi dello Stato, però, è abbastanza probabile che la chiave segreta del PGP risulti rapidamente compromessa e che dunque risultino molto più convenienti altre metodologie di attacco, basate sullo sfruttamento di debolezze umane e non più del sistema crittografico in sé.

Cerchiamo prima di tutto di capire meglio la funzione della passphrase. La sicurezza del PGP è data da due elementi che dovrebbero sempre rimanere segreti: la passphrase e la chiave segreta vera e propria. La chiave segreta è

costituita da un blocco di bit: è un piccolo file che viene custodito sul proprio computer o su un dischetto. Per maggior sicurezza per poter utilizzare propriamente la chiave segreta è anche necessario digitare una passphrase, cioè una parola d'ordine che può essere composta da parole, frasi o numeri di qualunque lunghezza. Chiave segreta e passphrase sono entrambe ugualmente necessarie per il processo di decodifica dei messaggi.

Si può ora immaginare una situazione tipica di attacco molto concreto al proprio sistema crittografico: il proprio computer cade completamente nelle mani del nemico (ad esempio a causa di un'improvvisa perquisizione). In questo caso, a meno che non si siano attuate ulteriori precauzioni (ad esempio l'uso di file system crittati o steganografati, come vedremo più avanti), la chiave segreta risulta compromessa e lo è di conseguenza il 50% della nostra sicurezza. Ma se abbiamo predisposto tutto con cura, il 50% di una quantità "relativamente infinita" rimane qualcosa di "relativamente infinito". L'importante è dunque proprio aver scelto bene la passphrase.

In casi del genere, infatti, il nemico avrà comunque bisogno della passphrase per riuscire a compromettere definitivamente l'intero sistema. Per ottenerla ha due strade: chiedercela direttamente, oppure trovarla da solo. Nel primo caso è utile notare che una simile passphrase, scelta necessariamente in base a criteri di lunghezza e complessità, è qualcosa che si può dimenticare molto facilmente; tra l'altro la legge italiana consente all'imputato di non rispondere alle domande, da qualunque parte provengano (questo è uno dei motivi per cui in certi casi è importante informarsi immediatamente se si è imputati o solo testimoni e tacere fino a quando non si ottiene una risposta certa raggiunta con l'aiuto di un avvocato). Gli aspetti giuridici di una simile situazione sono ancora molto incerti e in Italia vi sono pochissimi precedenti significativi. Può essere comunque interessante riportare che nei pochi casi in cui sono stati sequestrati dischi e computer contenenti anche (tra le altre cose) materiale crittato, gli investigatori non hanno fatto finora particolari pressioni per ottenere le chiavi di accesso.

Nel caso invece che il nemico decida, o sia costretto, a trovare la passphrase da solo, allora averla scelta bene manifesterà tutta la sua importanza. In rete sono disponibili alcuni piccoli programmi in grado di tentare numerose combinazioni al secondo. Coadiuvati da appositi dizionari elettronici, questi programmi possono ad esempio essere impiegati per "testare" tutte le parole più comuni della lingua italiana o di qualsiasi altra lingua conosciuta. È quindi fondamentale che la passphrase sia costituita da più parole, possibilmente inesistenti su qualsiasi vocabolario, comprendenti anche lettere maiuscole, numeri e caratteri di punteggiatura.

La passphrase non deve inoltre avere alcun riferimento a elementi pubblici o privati della propria vita o di quella dei propri amici o conoscenti: le prime centinaia di tentativi di un ipotetico attacco riguarderanno possibili date di nascita, numeri di codice fiscale, numeri di telefono, indirizzi, nomi scritti correttamente o rovesciati. Affidarsi a questo tipo di dati per scegliere la propria passphrase è una pessima idea. Se ci sono dei dubbi in proposito o se la paranoia sembra eccessiva, basta ricordare che la crittografia è un'arte molto antica e che sono già stati accumulati un paio di millenni di esperienza sui modi più comuni usati dall'uomo per scegliere e memorizzare codici segreti.

Sono stati consigliati vari modi pratici e sicuri per arrivare a scegliere una passphrase sufficientemente robusta. Uno dei più semplici e comuni è il seguente: prima di tutto si costruisce una frase abbastanza lunga ma facile da memorizzare. Ad esempio: "Accipicchia! Questo libro contiene 250 pagine di dinamite, gli amici di Superman non saranno contenti di vederlo…".

La nostra passphrase sarà quindi costituita semplicemente dalle iniziali di ogni singola parola della frase, compresa la punteggiatura:

"A!Qlc250pdd¬ga5858dSnscdv…"

È facile immaginare come questo sia comunque solo uno dei molti modi (e nemmeno il più sicuro) per costruire una passphrase a prova di qualsiasi scanner.

La passphrase ideale è costituita da una sequenza di caratteri perfettamente casuali. Il requisito della perfetta casualità, apparentemente banale, in realtà rappresenta un problema che pone diverse difficoltà. Nell'esempio appena suggerito, le lettere minuscole appaiono più frequentemente di quelle maiuscole e le consonanti più frequentemente delle vocali (nella lingua italiana le vocali appaiono più spesso come finale di sillaba, tendendo quindi a non comparire all'inizio di parola). Questo tipo di debolezze, se opportunamente analizzato e sfruttato, può facilitare un tentativo di attacco. La casualità perfetta, o almeno una sua accettabile approssimazione, in ambito informatico si raggiunge spesso affidandosi all'ambiente fisico esterno al computer: nel momento in cui necessitano di una serie di numeri casuali, alcuni programmi chiedono all'utente di spostare il mouse a caso, oppure di campionare il rumore di fondo presente nella stanza attraverso la scheda audio; il movimento del mouse e il rumore di fondo verranno poi convertiti in sequenze di informazioni e quindi in numeri. Il PGP, dal canto suo, al momento della generazione della prima coppia di chiavi chiede all'utente di digitare alcune lettere a caso sulla tastiera del computer: ad essere interpretati e convertiti in numeri, dopo adeguate trasformazioni, saranno in questo caso gli intervalli di tempo trascorsi tra la pressione di un tasto e l'altro.

Lo *scanning* sistematico attraverso appositi programmi costituisce però solo una delle molte strategie possibili per ottenere la passphrase senza la nostra collaborazione (stiamo comunque sempre ipotizzando che la sicurezza della chiave segreta sia già stata compromessa). Altri metodi possono risultare tanto banali quanto efficaci: se il proprio computer è posto vicino a una finestra, come spesso accade, un buon telescopio piazzato per un congruo periodo di tempo nel condominio di fronte può permettere di leggere tutti i movimenti delle nostre dita sulla tastiera. Per non parlare, ovviamente, di eventuali altre persone presenti alle nostre spalle mentre digitiamo la passphrase.

Altre strategie sono meno banali e talvolta sorprendono gli utenti con minore familiarità nei confronti dei mezzi informatici. È il caso di una famiglia di programmi denominati "keyloggers", o anche (se utilizzati in rete) "sniffers". Questi piccoli programmini, preventivamente installati in un computer, provvedono a registrare a sua insaputa tutte le operazioni svolte dall'utente, comprese le sequenze di tasti digitati. In via teorica è possibile penetrare di nascosto in casa altrui, modificare la configurazione software del computer e fare in modo che, agli avvii successivi, il keylogger venga caricato automaticamente. Dopo un certo periodo di tempo, in un ulteriore sopralluogo si provvederà a prelevare il keylogger e tutto il suo bagaglio di informazioni. Non è sempre facilissimo, nemmeno per gli utenti più smaliziati, rendersi conto di avere un simile intruso nel proprio computer. La soluzione è quella di non lasciare mai il proprio pc eccessivamente incustodito (tenendo anche presente che l'eventuale password del BIOS impostata all'avvio si supera abbastanza facilmente aprendo fisicamente il cabinet e spostando un *jumper* all'interno) e soprattutto verificare spesso che la configurazione di avvio non sia cambiata. Quest'ultimo controllo potrebbe probabilmente essere reso automatico e molto più preciso utilizzando qualche programma predisposto *ad hoc*.

Un'ulteriore possibilità di attacco riguarda gli utenti di Windows e di tutti quei sistemi operativi che fanno uso della memoria virtuale su disco. In questi sistemi, parte della memoria di lavoro (che contiene anche dati e comandi immessi da tastiera, compresa l'eventuale passphrase) viene temporaneamente trasferita sul disco fisso, in un file chiamato file di *swap* (o *swap file*). Queste informazioni possono rimanere nello *swap file* anche dopo il termine della sessione di lavoro e lo spegnimento del computer. Per essere sicuri che la passphrase (o altre informazioni delicate) non sia rimasta da qualche parte scritta su disco, è necessario cancellare sistematicamente il contenuto dello *swap file*. Quello che ancora molti non sanno è che il comune comando usato per cancellare i file in realtà non effettua una reale cancellazione fisica e rende possibile (e anche molto semplice) il loro recupero in tempi successivi. Sarà

quindi indispensabile che lo *swap file* venga cancellato *in modo sicuro*, attraverso le numerose utility specificamente disponibili a questo scopo. Alcune di queste possono anche essere configurate in modo che lo *swap file* venga automaticamente e irrimediabilmente cancellato alla fine di ogni sessione di lavoro. Sul problema della cancellazione *sicura* dei file dal disco fisso, comunque, si tornerà nel capitolo dedicato ai *file system crittati*.

La metodologia di attacco di cui ultimamente si è più parlato in certi ambienti è il cosiddetto attacco TEMPEST, o attacco di Van Eck. Molte apparecchiature elettroniche, tra cui i comuni monitor per personal computer, emettono onde elettromagnetiche. Queste onde possono essere ricevute e decodificate a distanza, con appositi strumenti, fino a permettere, nel caso di un monitor, la riproduzione di ciò che appare a video. Il messaggio privato che si crede di leggere sul proprio monitor nell'intimità della propria casa può apparire sullo schermo nemico grazie a speciali apparecchiature poste nella stanza, o nell'appartamento accanto. Questa tipologia di attacco, a causa della sua natura completamente passiva (il nemico si limita ad "ascoltare" le onde elettromagnetiche emesse dal nostro monitor, senza manifestare la sua presenza in alcun modo) semplicemente non può essere individuata.

Ci si può cautelare preventivamente da un attacco TEMPEST mediante schermatura del monitor, tenendo però presente che una schermatura completamente efficace richiederebbe accessori del costo di svariate decine di milioni di lire. Oppure, molto più semplicemente, si può provvedere a un adeguato disturbo delle emissioni del proprio monitor, affiancandogli un ulteriore apparecchio il cui unico scopo è appunto quello di emettere a sua volta altre onde elettromagnetiche che confonderanno l'ambiente. Apparecchi di questo tipo possono essere autocostruiti con poca spesa. In ogni caso l'emissione di onde elettromagnetiche è circoscritta a pochi metri, e quindi ogni tentativo di attacco nemico deve necessariamente posizionarsi molto vicino alla fonte delle emissioni. È solo il caso di notare, infine, che un attacco di questo tipo mirerebbe alla lettura dei messaggi così come noi li leggiamo sul nostro monitor, ma non otterrebbe comunque la passphrase, che non viene visualizzata a video mentre la si digita.

Infine, un potenziale pericolo proviene dal software di crittografia stesso. La copia del PGP che si sta utilizzando, ad esempio, potrebbe esser stata modificata preventivamente dal nemico, al fine di comportarsi in modo "anomalo": registrando la passphrase da qualche parte oppure inglobando una *backdoor* in grado di garantire a determinate persone l'accesso ai nostri dati. A questo tipo di minacce ci si riferisce con il nome di "cavalli di Troia". Per evitare di accogliere un simile "cavallo di Troia" nel proprio sistema sono sufficienti poche semplici precauzioni: in primo luogo, il software va sempre

prelevato da fonti sicure. Evitare nel modo più assoluto di servirsi ciecamente di programmi prelevati da siti sconosciuti o non ufficiali, oppure ricevuti da persone di cui non si nutre assoluta fiducia sia per quanto riguarda la loro integrità etica, sia per quanto riguarda le loro competenze tecniche (qualche sincero amico potrebbe passarci un "cavallo di Troia" in perfetta buona fede). In secondo luogo, ogni software di crittografia "serio" dovrebbe essere accompagnato da una firma digitale dell'autore che ne certifichi la non manomissione. Questa non è di per sé garanzia di buon funzionamento, ma la può diventare se quel programma diventa sufficientemente verificato e conosciuto: le migliaia di prove e verifiche riportate in rete riguarderanno una versione del software contrassegnata da una particolare firma; se le firme corrispondono, si può quantomeno essere certi di stare utilizzando esattamente lo stesso programma oggetto di queste verifiche, e non una sua versione contraffatta.

PGP: QUALE VERSIONE?

In conclusione di questo capitolo, è il caso di affrontare brevemente una questione che ha suscitato molte discussioni. Ci si riferisce alle varie versioni del PGP che si sono succedute soprattutto a partire dal 1993. Si proverà quindi a fare una rapida rassegna delle versioni principali, indicandone le differenze degne di nota.

PGP 2.3a – è la versione che ha fatto conoscere il PGP in tutto il mondo, sviluppata da Phil Zimmermann; alcune persone continuano a usarla, un po' per pigrizia e scarsa voglia di cambiare software, un po' per alcune voci che la ritengono l'unica versione assolutamente sicura. Con questa versione non è possibile leggere i messaggi generati con alcune delle versioni più recenti e non è possibile in ogni caso utilizzare chiavi maggiori di 1280 bit. Inoltre, questa versione utilizza una particolare libreria software (MPILIB) contenente algoritmi protetti da brevetto, che ha causato a Zimmermann alcune noie legali da parte della RSA, Inc. (per questo motivo l'utilizzo del PGP 2.3a è illegale all'interno degli USA, in quanto il brevetto è valido solo all'interno degli Stati Uniti).

PGP 2.6ui (le lettere "ui" significano "*unofficial international*" e indicano una versione compilata all'esterno degli USA, in genere per sottrarsi ai problemi legali e di brevetti di quel paese) – è una versione basata sul codice della 2.3a, ma a differenza di quest'ultima permette di comunicare liberamente con tutti gli utilizzatori delle versioni più recenti. A questo proposito ricordiamo che,

per sottrarsi alla causa legale intentata contro di lui dai titolari dei brevetti per l'algoritmo RSA, Phil Zimmermann ha acconsentito ad accogliere due precise condizioni: le versioni ufficiali sviluppate negli Stati Uniti successive alla 2.3a utilizzano una nuova libreria software (RSAREF) appositamente rilasciata dall'RSA per l'uso non commerciale, libera da royalty (ma dimostratasi altrettanto sicura) e, per incentivare l'utilizzo di queste nuove versioni, esse vengono rese artificiosamente incompatibili con i messaggi generati dalla 2.3a. Ribadiamo che queste decisioni sono state prese per problemi di brevetti e non per pressioni dell'FBI o di altre agenzie governative. In ogni caso, molti utilizzatori di tutto il mondo non hanno ritenuto giusto accettare passivamente queste condizioni, imposte comunque esclusivamente per i cittadini americani e si è assistito così a un proliferare di versioni internazionali "non ufficiali" più o meno affidabili. Il PGP 2.6ui è sicuro quanto la versione 2.3a, poiché utilizza praticamente lo stesso codice, ma si presenta all'esterno come una versione 2.6 permettendo appunto di aggirare lo scoglio legale della compatibilità all'indietro (a questo problema ci si riferisce in genere come "*legal kludge*"); per questo stesso motivo il suo utilizzo è probabilmente illegale all'interno degli USA.

PGP 2.6.2 – sviluppata dal MIT (Massachusetts Institute of Technology), rappresenta la svolta "legale" del PGP all'interno degli Stati Uniti; utilizza la libreria RSAREF, di libero impiego ed è per questo leggermente più lenta delle versioni internazionali e della 2.3a; a causa del *legal kludge* i messaggi generati con questa versione non possono essere letti dagli utilizzatori di PGP 2.3a. Come tutte le versioni del PGP, la sua esportazione dagli USA è assolutamente proibita; una volta esportata, comunque, il suo utilizzo è perfettamente legale in tutto il resto del mondo.

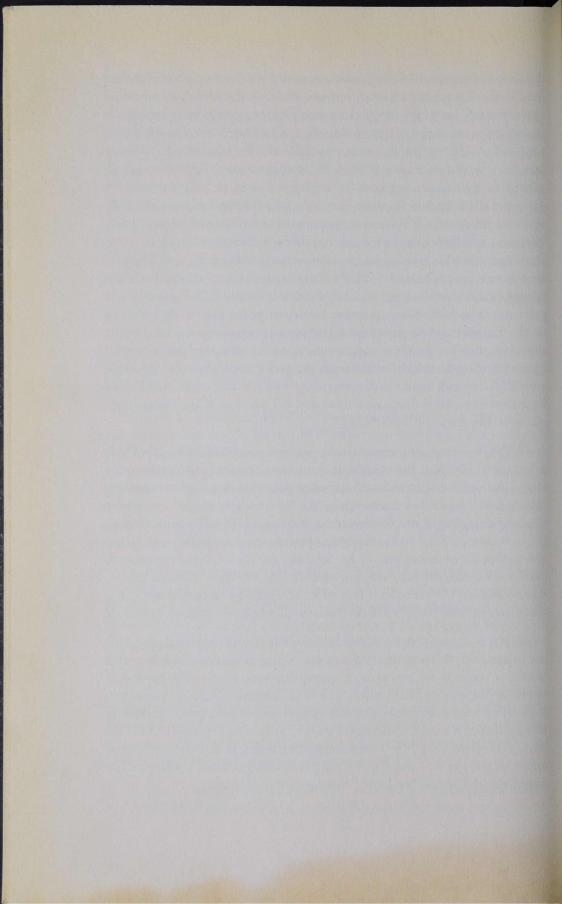
PGP 2.6.3i – basata sul codice della 2.6.2 del MIT e adattata per uso internazionale, ignora le limitazioni imposte alle versioni ufficiali americane: utilizza la libreria "proibita" MPILIB e può comunicare liberamente con le vecchie versioni (il suo utilizzo è di conseguenza da ritenersi illegale negli Stati Uniti). È probabilmente la versione della serie 2.6.x più efficiente e sicura ed è anche la versione consigliata a chiunque voglia iniziare a usare il PGP.

PGP 2.6.3 – il codice sorgente della 2.6.3i può essere compilato con un'opzione particolare per inglobare le limitazioni imposte negli USA (utilizzo della libreria RSAREF e incompatibilità con le versioni precedenti); l'unica caratteristica degna di nota di questa versione è quella di essere completamente legale negli Stati Uniti; non pensiamo che interesserà a molti.

PGP 5.0 – sviluppata dalla PGP, Inc., società fondata da Zimmermann nell'intento di sfruttare commercialmente la propria notorietà e tuttavia rapi-

damente venduta alla McAfee (famosa per i suoi prodotti antivirus). Costituisce l'inizio di una nuova serie di versioni del PGP, che però non ha lasciato tutti entusiasti. Le novità maggiori sono rappresentate da un'interfaccia grafica estremamente semplice e comoda nelle versioni per Windows e McIntosh e dall'utilizzo dell'algoritmo a chiave pubblica Diffie-Hellman, un sistema libero da copyright alternativo all'RSA. È distribuita in versione *freeware* (gratuita per l'uso individuale, seguendo la "tradizione") o in versione commerciale (destinata alle aziende e all'uso professionale). Le due versioni sono identiche quanto a sicurezza, ma la versione free genera esclusivamente chiavi Diffie-Hellman e non RSA: questa caratteristica, che ha lasciato perplessi molti utenti, rende necessaria l'importazione (possibile senza problemi) di chiavi RSA create con versioni precedenti. Il PGP 5.0 deve quindi essere affiancato da una versione della serie 2.x.x per chi si appresta a utilizzare il PGP per la prima volta e deve quindi ancora generare la propria prima coppia di chiavi. Le uniche due alternative sono quella di comprare la versione commerciale (che molti rifiutano per principio) oppure rassegnarsi a utilizzare per la propria chiave solo l'algoritmo Diffie-Hellman, che però è totalmente incompatibile con l'RSA e quindi consente di comunicare solo con altri utenti di PGP 5.0. Essendo una versione destinata al mercato USA, adempie al *legal kludge* e dunque non comunica con il PGP2.3a.

PGP 5.0i – versione internazionale, praticamente identica alla 5.0. Per la prima volta la versione internazionale non è stata ricavata da un'esportazione illegale del software bensì da un'estenuante scanning di codice stampato su libri di carta esportati in modo legale. Si è trattato di un modo per mettere ulteriormente in evidenza l'insensatezza delle leggi USA sull'esportazione di prodotti crittografici. L'utilità di questa operazione e il suo indiretto richiamo alla legalità come valore non ci ha del tutto convinti. Ulteriore motivo di perplessità è dato dal fatto che, pur essendo una versione "internazionale", contrariamente alle precedenti non offre la possibilità di disabilitare il *legal kludge* e quindi non permette di comunicare liberamente con tutte le versioni.

PGP 5.5 – a partire dalla release 5.0 il PGP tenta di integrarsi il più possibile in pacchetti software destinati a essere impiegati in aziende e ambienti di lavoro; la versione 5.5 prosegue in questa direzione e non offre particolari motivi di interesse per l'uso privato. Le varie, successive versioni proliferate negli ultimi mesi sono probabilmente troppo recenti per essere considerate sufficientemente sicure. Questo scenario sta comunque cambiando molto in fretta e le versioni del PGP costituiscono argomenti di discussione sempre molto accesi. Per una panoramica più precisa e aggiornata si rimanda alle pagine web di questo libro (http://www.ecn.org/kriptonite).

Ecco. Ora cominciavo ad avercela un'idea di dove voleva arrivare Luthor...

Crittografia.... pure questo è un parolone, dici? Forse, ma in fondo è solo un modo per farti i fatti tuoi assieme a qualcun altro senza che nessuno ci possa mettere il naso. Interessante, no? Pensaci. Puoi dirti di tutto senza bisogno di farti vedere insieme. Senza bisogno di vedersi MAI, se vuoi. Senza che l'altro sappia chi sei davvero, addirittura.... Crittografia, criptografia... mi suona bene sulla lingua 'sto parolone, amico... Un po' come Kriptonite, eh? Quel minerale che se Superman solo l'annusa comincia a vomitare e a cagarsi, c'hai presente? Certo che è furbo Lex!

Capisci amico? Lui aveva messo me e altri ragazzi a lavorare su questi cosi, sui computer. Giusto quel tanto che serviva a saperli usare un po', niente di più. Ma di queste cose non ci aveva mai fatto sapere nulla. Per essere sicuro che nessuno capisse che aveva intenzione di usare questo sistema? Chissà. Il vecchio Lex ha una mente più contorta delle spire di un serpente, amico, e mica è facile contarle tutte quelle spire. Anche un po' di paura forse. Lex vuole che nessuno dei ragazzi possa saperne più di lui. O possa far qualcosa senza che lui lo sappia... Interessante, te l'ho detto. Con questo sistema, capisci, se qualcuno mi beccava, potevo pure ridergli in faccia. Perché tanto gli sbirri non potevano capire quello che avevo nel computer. A meno che uno non fosse così idiota da dargli la sua parola d'ordine. E amico, non faccio per vantarmi, ma far sputare qualcosa al qui presente non è mica tanto facile. Ci hanno provato in tanti, e in tanti modi. Ma ancora nessuno può vantarsi di esserci riuscito... Ci credo che allo zio Sam non gli sfagiola troppo questa cosa del PGP. Come dici? Se ci credo davvero che nemmeno la CIA e l'NSA possono decifrare questo PGP? Bah amico, sicuro non lo sono. Tutti quei discorsi lì, la "potenza di calcolo", gli "algoritmi" o come cazzo si chiamano quelle robe, insomma non è mica pane per i miei denti, quello. Ma se va bene a 'sti cervelloni che ci passano la vita a studiarla questa roba, beh, allora può andar bene anche a me, ti pare? Anche perché, credimi, il mondo è pieno di idioti convinti che il vecchio zio Sam e i suoi nipotini non li potrai mai far fessi. Ma si

sbagliano. Perché non sono così invincibili, non sanno tutto di tutti come ti vogliono far credere. Parola di Joe Lametta, non sarei ancora in circolazione sennò... Ma nel frattempo facevo meglio ad occuparmi di un po' dei fatti miei. Lex era in attesa del mio messaggio, dovevo dirgli che la bomba era sistemata e che tutto era filato liscio. E non conveniva farlo aspettare troppo. Le finestre del cottage erano già tutte chiuse. È la prima cosa da fare appena entrati, ormai è come un'abitudine. Se non c'hai di queste abitudini prima o poi ti trovi la rogna di dover accoppare qualche povero stronzo di guardone, non so se mi spiego. Al e Louie erano nell'altra stanza a giocare a poker e a scambiarsi insulti. E loro lo sanno che è poco igienico venire a ficcanasarmi alle spalle mentre faccio i fatti miei. Unica cosa, meglio schermare quel fottuto monitor, no? Non pensavo che ci fosse un furgone pieno di sbirri e zeppo di fottute apparecchiature di ascolto parcheggiato all'angolo, ma coi SuperSensi dell'Omino d'Acciaio meglio andare sul sicuro, ci aveva pensato anche Lex, uno degli aggeggi nella valigetta serviva proprio a quello. Ero pronto. Solo, dove cazzo lo trovavo 'sto fottuto PGP? E come funzionava?

Primi passi con il PGP

di T.H.E Walrus

Questo testo è il riadattamento di una guida scritta per essere consultata on-line e pensata per aiutare i principianti a installare il PGP e familiarizzare con esso. Essendo una guida per chi è ancora all'oscuro di tutto, offre una conoscenza tutt'altro che completa. Non ci si stancherà mai di ricordare che, per un uso sicuro e consapevole di PGP, è *indispensabile una conoscenza approfondita* della documentazione acclusa al software. In rete sono consultabili altri strumenti: basti ricordare il manuale di Zimmermann incluso nel pacchetto di distribuzione del PGP, di cui esiste anche una versione italiana curata da Marco Giaiotto, oppure la guida introduttiva in italiano di Giorgio Chinnici.

Dove trovare il PGP e quale versione scegliere

La storia del PGP e dei tentativi fatti dai vari governi per impedirne o limitarne la diffusione è lunga e complessa. Nell'introduzione di questo libro sono state delineate le vicende storiche che hanno caratterizzato la nascita di questo software; nel presente capitolo ci si limiterà a sottolineare che a causa di vari motivi - il divieto di esportazione di software di crittografia forte dagli USA, le restrizioni commerciali poste all'impiego dell'algoritmo di crittazione RSA utilizzato da tutte le versioni meno recenti di PGP per la generazione delle chiavi, ecc. - sono proliferate numerose versioni di PGP, la cui effettiva sicurezza è a volte oggetto di dubbio. Senza entrare in dettaglio, basterà dire

che per tutti coloro che *non* risiedono negli USA è universalmente considerata sicura - in base a criteri altrove esposti - oltre che legale, la versione 2.6.3i ("i" sta per internazionale) di Stale Schumacher. Si può rintracciare questa versione - oltre che su http://www.ecn.org/crypto - anche nell'home page internazionale del PGP: http://www.pgpi.com. Questo è il sito più importante per il PGP: contiene tutte le versioni di PGP disponibili per i vari sistemi operativi, più collegamenti, documentazione e una serie di informazioni di vario tipo estremamente utili.

Le versioni più recenti del PGP (a partire dalla 5.0) si differenziano dalle precedenti per due aspetti fondamentali. In primo luogo il programma non funziona più su riga di comando, ma si integra alle interfacce grafiche di diffusi sistemi operativi quali Windows 95/98, NT e Macintosh. In secondo luogo vengono utilizzati algoritmi di crittazione alternativi - come il Diffie-Hellman - in sostituzione del noto RSA utilizzabile solo nella versione a pagamento per la generazione e la gestione delle chiavi.

Sul PGP 5.0 ci sono state - e sono tuttora in corso - molte discussioni. Non sembra legittimo considerare "inferiore" la sicurezza offerta da questo sistema, se si tiene presente che anche per esso i sorgenti sono stati resi pubblici, sia pur in maniera inconsueta, attraverso la scansione del libro dove erano pubblicati. Non essendo il libro soggetto a restrizioni di esportazione dagli USA, a partire dalla copia di tali sorgenti Schumacher e collaboratori hanno costruito una versione "internazionale" analoga a quella americana, chiamata PGP 5.0i. Le voci che sono circolate su presunte *backdoor* di quest'ultima release sono in effetti dovute alla presenza di un'ulteriore versione - la 5.5, destinata al mercato delle aziende - che permette (se configurata opportunamente) all'amministratore di sistema la lettura di tutte le lettere crittate dagli impiegati.

Persistono tuttavia alcune perplessità sull'uso del PGP 5.0. A parte il taglio decisamente più commerciale, in questa versione ci sono problemi di compatibilità con le precedenti release: ad esempio, le chiavi create con la nuova versione *freeware* sono illeggibili per tutte le vecchie versioni (e quindi permettono di comunicare soltanto fra persone che utilizzano il PGP 5.0). In secondo luogo la scelta dell'algoritmo Diffie-Hellman (a detta di molti esperti) sarebbe dovuta non alla ricerca di maggiore sicurezza, ma alla ricerca di un algoritmo non brevettato e di pubblico dominio, libero quindi dal pagamento delle royalty.

La perplessità più grande, riguarda però il fatto che l'integrazione all'interfaccia grafica del sistema operativo, per quanto comodissima, è del tutto trasparente e conduce l'utilizzatore ad un uso puramente meccanico di

PGP, senza stimolare una comprensione sostanziale di ciò che sta facendo e del funzionamento del programma. La piena capacità di valutare questi elementi risulta invece fondamentale per padroneggiare in modo realmente sicuro il PGP.

Per questo motivo è meglio avvicinarsi al PGP nella maniera più "dura", magari familiarizzando con la versione 2.6.3i, più spartana e complessa, ma infinitamente più lucida ed evidente. Anche se si utilizza Windows, la 2.6.3i gira benissimo in shell DOS e la scomodità iniziale di dover lavorare dal prompt del DOS è ampiamente compensata dalla migliore comprensione delle modalità di funzionamento del PGP. Quando ci si sentirà in grado di padroneggiare sufficientemente lo strumento, sarà il momento di valutare se conviene o meno passare a versioni successive come la 5.0.

La versione "ufficiale" del PGP 2.6.3i viene distribuita in un file chiamato pgp263i.zip: è la versione base per DOS di Stale Schumacher. Esistono anche altri pacchetti "ufficiali" di questa release con nomi leggermente diversi, come una versione per DOS a 32 bit, versioni per altri sistemi operativi o i file sorgenti.

Dato che esiste sempre il rischio (quello cypherpunk è un mondo giustamente paranoico) che qualcuno stia diffondendo una versione manipolata e insicura del PGP, dovrebbe essere sempre possibile controllare l'integrità del pacchetto pgp263i.zip usando il certificato di firma pgp263ii.asc compreso nel pacchetto di distribuzione, appositamente rilasciato da Schumacher.

Installazione

È tempo dunque di gettare uno sguardo alla parte tecnica. Una volta procurato il file pgp263i.zip, occorre procurarsi il programma di decompressione (pkzip/pkunzip) e creare sul proprio disco una directory apposita in cui decomprimere il PGP. Chiameremo per esemplificare questa directory c:\pgp. Eseguita la decompressione, in c:\pgp si dovranno trovare i seguenti file:

```
pgp263ii.asc
pgp263ii.zip
readme.1st
readme.usa
setup.doc
```

Concentriamoci ora sulla presenza di un ulteriore pacchetto "zippato" da decomprimere (pgp263ii.zip) che contiene i seguenti file:

```
config.txt - de.hlp - en.hlp - es.hlp - fr.hlp - keys.asc -
language.txt - md5sum.exe - pgp.exe - pgp.hlp - pgpsort.exe
```

pgp263ii.zip va decompresso usando il comando <pkunzip -d> (o qualsiasi altra opzione del vostro decompressore che ricrei le subdirectory) in modo da creare automaticamente la subdirectory c:\pgp\doc, dove verrà archiviata tutta la documentazione. Bisogna modificare con un qualunque editor ASCII (ad esempio edit.com del DOS o Notepad di Windows) il proprio c:\autoexec.bat, inserendo dopo il path le seguenti *esatte* stringhe:

```
SET PGPPATH=C:\PGP
SET PATH=C:\PGP;%PATH%
SET TZ=MET-1DST
```

La prima stringa setta la variabile d'ambiente pgppath, la seconda integra il path del PGP al path preesistente; queste due stringhe sono *necessarie* (ovviamente, se il PGP è stato installato in un'altra directory bisogna indicare il percorso giusto). La terza stringa serve a settare la *timezone*, in modo che ogni operazione fatta con il PGP rechi l'ora e la data corretta. Salvate le modifiche all'autoexec.bat, occorre aprire con il solito editor di testo il file config.txt che si trova in c:\pgp. Ci si troverà dentro un sacco di roba, che servirà più avanti a configurare il PGP in modo più "evoluto". Per ora non va modificato quasi nulla. Bisogna cercare la seguente stringa:

```
# Legal_Kludge = off
```

ed eliminare il cancelletto, ottenendo questo risultato:

```
Legal_Kludge = off
```

Questa operazione serve a rendere le chiavi generate con questa versione del PGP *compatibili* anche con le versioni precedenti. Tutte le informazioni sul *legal kludge* e sul motivo per cui in USA la compatibilità è diventata un problema legale, hanno il sapore di una telenovela (infinita) e sono gustosamente reperibili (per chi lo desideri) nella documentazione compresa nel pacchetto.

Salvata la modifica al config.txt di PGP, è necessario riavviare il proprio PC per rendere operative le modifiche apportate all'autoexec.bat. Sarà poi utile creare una directory tipo c:\pgp\prova, da utilizzare come campo di battaglia per i successivi tentativi di familiarizzazione con il programma. Il primo comando da battere stando dentro c:\pgp\prova, è:

```
pgp -h
```

Se tutti i passi finora indicati sono stati seguiti correttamente, compariranno le schermate d'aiuto del PGP. Poiché questo è il comando che si usa *più spesso* per ricordare i vari parametri e la loro sintassi, potrebbe non essere una cattiva idea stampare le schermate in modo da averle sott'occhio facilmente in ogni momento. A questo punto l'installazione è finita. Si può uscire dalla schermata d'aiuto e passare alla generazione della propria coppia di chiavi.

GENERARE LA PROPRIA COPPIA DI CHIAVI

Le *chiavi* sono il cardine intorno al quale si impernia tutto il sistema della crittazione PGP. Una volta terminata la generazione si avrà la propria *chiave pubblica* con cui qualunque persona potrà crittare qualunque file desideri (file binari, di testo, eseguibili, eccetera). Soltanto chi avrà accesso alla *chiave privata corrispondente* (e non si insisterà mai abbastanza sul fatto che *solo tu* dovresti poter accedere alla *tua* chiave privata) potrà poi eseguire la decrittazione. Questo discorso può apparire ora illogico o complicato, ma proseguendo nella lettura si scoprirà che non ci vuole molto a capire come si fa.

La chiave pubblica e la chiave privata corrispondente vengono generate insieme, *con un'unica operazione*. Digitando:

```
pgp -kg
```

comparirà la seguente schermata, con cui PGP chiede di scegliere la dimensione (cioè il numero di bit) della coppia di chiavi che si vuol generare:

```
*********************************************************************
Pretty Good Privacy(tm) 2.6.3i - Public-key encryption for the masses.
(c)1990-96 Philip Zimmermann, Phil's Pretty Good Software. 1996-01-18
International version - not for use in the USA. Does not use RSAREF.
Current time: 1998/10/28 20:31 GMT

Pick your RSA key size:
1) 512 bits- Low commercial grade, fast but less secure
2) 768 bits- High commercial grade, medium speed, good security
3) 1024 bits- "Military" grade, slow, highest security

Choose 1, 2, or 3, or enter desired number of bits:
*********************************************************************
```

Fra le tre opzioni proposte (512, 768 e 1024 bit) è consigliabile la chiave più grande: da 1024 bit. Chiavi più piccoline danno minore sicurezza, mentre una

chiavona esagerata (sino a 2048 bit) per ora non offre particolari vantaggi in termini di sicurezza e in più non è compatibile con il PGP versione 2.3a. Per cui si sceglie l'opzione 3 e si vedrà comparire la seguente schermata:

```
*********************************************************************
Generating an RSA key with a 1024-bit modulus.

You need a user ID for your public key. The desired form for this
user ID is your name, followed by your E-mail address enclosed in
<angle brackets>, if you have an E-mail address.
For example: John Q. Smith <12345.6789@compuserve.com>

Enter a user ID for your public key:
*******************************************************************
```

Il PGP sta chiedendo un nome da associare alla coppia di chiavi (*User ID*), che può essere costituito da una o più parole e caratteri alfanumerici. Il PGP userà questo User ID per identificare le tue chiavi durante la crittazione\decrittazione dei messaggi[1].

È possibile fare in modo che compaia, a fianco del proprio User ID, qualche altra informazione, come ad esempio un indirizzo di e-mail. In questo caso occorre digitare le informazioni aggiuntive tra i simboli < >, ma sebbene sia una pratica molto diffusa se ne può anche fare a meno o si può attuare in seguito. Per esempio, se il tuo alias è "Lametta" puoi limitarti a battere

```
Lametta
```

Oppure, se vuoi comprendere nello User ID anche gli indirizzi di posta elettronica che hai eventualmente a disposizione:

```
Lametta <joe@luthorcorp.com>
```

Dopo aver scelto lo User ID, il PGP ti avvertirà di scegliere la tua *passphrase*.

```
*****************************************************************
You need a pass phrase to protect your RSA secret key.
Your pass phrase can be any sentence or phrase and may have many
words, spaces, punctuation, or any other printable characters.

Enter pass phrase:
*******************************************************************
```

[1] È più semplice utilizzare inizialmente come user ID il nome o l'alias con cui si è più conosciuti in rete. Ovviamente nulla vieta in seguito di cambiare lo user ID o di aggiungerne altri alla propria coppia di chiavi o ancora di creare tante coppie di chiavi quante sono le identità che si assumono in rete.

Il PGP chiede di digitare una frase o parola (la *passphrase*, appunto), senza la quale sarà impossibile accedere alla tua chiave privata. Qualora qualcuno entrasse in possesso della tua chiave privata (che è comunque bene che non vada *mai* in giro) non potrà ugualmente decrittare i messaggi crittati con la tua chiave pubblica, a meno che non conosca anche la passphrase.

Per la passphrase puoi utilizzare tutte le lettere, numeri, spazi e segni di punteggiatura che puoi battere. Cerca un compromesso tra una frase abbastanza facile da ricordare (non dovrebbe *mai* essere scritta da nessuna parte) e abbastanza strana da non poter essere facilmente immaginata (scegliere come passphrase "tanto va la gatta al lardo..." o il nome di battesimo del tuo amore, o la data di nascita di tuo figlio è l'equivalente di scegliere "pippo" come User ID...). Probabilmente la miglior soluzione è rappresentata da una frase facilmente memorizzabile da parte tua, anche se non facilmente collegabile a te da chi ti conosce, inframmezzata da caratteri numerici e simboli[2]. Batti la passphrase che hai scelto e vedrai comparire:

```
*****************************
Enter same pass phrase again:
*****************************
```

Qui bisogna ribattere la passphrase scelta per conferma (se si sbaglia a batterla verrà chiesto di reinserirla e riconfermare da capo). Comparirà infine:

```
************************************************************************
Note that key generation is a lengthy process.

We need to generate 824 random bits. This is done by measuring the
time intervals between your keystrokes. Please enter some random text
on your keyboard until you hear the beep:

824
************************************************************************
```

Il PGP sta chiedendo di battere a caso sulla tastiera un certo numero di volte per introdurre un ulteriore elemento di casualità nella generazione delle chiavi. Se obbedisci a quest'ultima richiesta, il PGP ti ringrazierà cortesemente:

```
****************************
* -Enough, thank you.
****************************
```

[2] Bisogna tenere presente che il PGP nella digitazione della passphrase è sensibile alla differenza fra maiuscole e minuscole. Se ad esempio si sceglie come passphrase "yuk yuk", quando poi il PGP chiederà di digitare la propria passphrase per decrittare un messaggio, se si digiterà "Yuk yuk" si otterrà solo il fatidico messaggio: "Error: Bad pass phrase".

E poi, mentre rumina per costruire le chiavi farà scorrere 'sta roba:

```
............****.........****.........****.........****.....
```

Ci può volere un po' di tempo, a seconda della potenza della macchina. Alla fine si vedrà comparire:

```
****************************************
Pass phrase is good. Just a moment....
Key signature certificate added.
Key generation completed.
****************************************
```

E le chiavi sono pronte.

Ora che ho le chiavi, come le devo usare?

Mentre ruminava, il PGP ha creato le chiavi e due file particolari, i cosiddetti "mazzi di chiavi" (*keyring*) nei quali conserva le chiavi stesse. Se si sono seguite le indicazioni finora offerte, i due *keyring* si troveranno in c:\pgp. Si chiameranno pubring.pgp (che contiene le chiavi pubbliche) e secring.pgp (che contiene le chiavi private).

Per ottenere la propria chiave pubblica da mandare in giro per il mondo, bisognerà estrarla dal pubring.pgp (vedremo più tardi come), esattamente come per consegnare ad un amico la chiave di casa bisogna sfilarla dal mazzo di chiavi. La chiave privata invece va conservata dentro il secring.pgp e serve a decifrare i messaggi che le persone che desiderano comunicare privatamente con noi critteranno con la nostra chiave pubblica[3]. Per ora nei keyring ci sono solo le due chiavi appena generate. Per visualizzarle bisogna digitare il comando:

```
pgp -kv
```

[3] Un consiglio da amico contro la sfiga pura è: *salva i keyring su dischetto* prima di mandare in giro la tua chiave pubblica. E dopo che li hai salvati, ricorda di salvarli di nuovo periodicamente, perché così:
a) quando prima o poi qualcosa andrà a puttane sull'hard disk e toccherà reinstallare anche il PGP, non dovrai far altro che sovrascrivere i nuovi keyring con quelli salvati su dischetto. In questo modo non hai bisogno di doverti fare una nuova coppia di chiavi, che è sempre un po' un casino (consulta la documentazione per la procedura di *revoca* di una chiave) e non dovrai cercare o farti rimandare di nuovo le chiavi pubbliche nel frattempo accumulate, che è sempre un po' una rottura per te e per il prossimo.

comparirà una scritta tipo:

```
**********************************************************************
Pretty Good Privacy(tm) 2.6.3i - Public-key encryption for the masses.
(c) 1990-96 Philip Zimmermann, Phil's Pretty Good Software. 1996-01-18
International version - not for use in the USA. Does not use RSAREF.
Current time: 1998/10/28 20:41 GMT

Key ring: 'c:\pgp\pubring.pgp'
Type   Bits/   KeyID      Date        User ID
pub    1024/   C4E35621   1998/10/28   Lametta <joe@luthorcorp.com>
1 matching key found.
**********************************************************************
```

Nelle ultime quattro righe, il PGP scrive da quale keyring sta visualizzando la chiave (in questo caso da c:\pgp\pubring.pgp) e poi dà una serie di informazioni, cioè:

- tipo di chiave (*pub*, per pubblica)
- numero di bit della chiave (*1024*)
- keyID (*identificatore alfanumerico della chiave*)
- *data* in cui la chiave è stata generata
- *User ID* completo di eventuali ulteriori informazioni.

Il comando <pgp -kv>, battuto senza indicare il path del keyring, visualizza di default le chiavi pubbliche contenute nel pubring.pgp, in qualunque directory si stia lavorando. Per visualizzare le chiavi private contenute nel secring.pgp, si usa lo stesso comando, aggiungendo però il percorso completo (*full path*) nel quale si trova il secring, in questa maniera:

```
pgp -kv c:\pgp\secring.pgp
```

Si ottiene così questa schermata:

```
**********************************************************************
Pretty Good Privacy(tm) 2.6.3i - Public-key encryption for the masses.
(c) 1990-96 Philip Zimmermann, Phil's Pretty Good Software. 1996-01-18
International version - not for use in the USA. Does not use RSAREF.
Current time: 1998/10/28 21:58 GMT
```

b) se giri per il mondo e stai a casa di un amico che ha un PC e il PGP, puoi usare le tue chiavi dal dischetto senza dover reinstallare nulla (vedi comunque la documentazione del PGP a proposito). Naturalmente bisogna fidarsi abbastanza dell'amico ed è meglio evitare, in generale, di battere la propria passphrase in ambienti non sicuri, come ad esempio le reti locali delle università o delle ditte, dove possono girare più o meno legalmente dei programmi chiamati "sniffer" che leggono tutto quello che passa attraverso la rete, fra cui anche la tua chiave e la tua passphrase. Ma questo rientra in considerazioni più generali di sicurezza...

```
Key ring: 'c:\pgp\secring.pgp'
Type   Bits/  KeyID     Date        User ID
sec     1024/  C4E35621  1998/10/28   Lametta <joe@luthorcorp.com>
1 matching key found.
******************************************************************************
```

Stavolta il PGP informa che il keyring visualizzato è il secring.pgp e la chiave è segreta (*sec*). A parte questo, gli altri dati (compreso KeyID e User ID) sono uguali. Verificato che le chiavi sono a posto nei loro keyring, passiamo finalmente a crittare un file.

Crittazione di un file

Il PGP può produrre file crittati direttamente in binario, oppure li può rivestire di una "armatura di trasporto" (*armor*) che rappresenta i dati binari in caratteri ASCII stampabili. Questa seconda opzione è estremamente importante, perché il formato ASCII va utilizzato per tutto ciò che si vuol trasmettere via posta elettronica (messaggi, chiavi pubbliche, file) in quanto i dati binari potrebbero subire alterazioni durante il transito sui vari sistemi, divenendo illeggibili o indecodificabili.

Per semplificare proviamo però prima a crittare un file in formato binario. Questo formato è utile per tutto ciò che si vuole nascondere a occhi estranei, siano file di testo o file eseguibili, purché non li si voglia spedire nel corpo di un messaggio o di una e-mail. È essenzialmente la migliore soluzione per ciò che si vuole conservare crittato su disco o dischetto[4] o che si desidera spedire in *file attach*, perché "pesa" di meno del suo corrispondente ASCII. Creiamo dunque un piccolo file di prova, con un editor di testo (bastano un paio di righe) e salviamolo con il nome prova.txt, nella directory c:\pgp\prova. Per crittare si usa il comando:

```
pgp -e prova.txt
```

e si ottiene:

[4] Per crittare ciò che si vuole tenere riservato sul proprio hard disk o archiviare "solo per i propri occhi", è consigliabile l'uso dell'opzione <pgp -c> che non implica l'utilizzo della doppia chiave ma solo di una passphrase. Si occupa così anche in assoluto il minimo spazio su disco ed è probabilmente il tipo di crittazione più invulnerabile consentita dal PGP. Come al solito, per maggiori informazioni è meglio leggere la documentazione che accompagna il programma.

```
***********************************************************************
Pretty Good Privacy(tm) 2.6.3i - Public-key encryption for the masses.
(c) 1990-96 Philip Zimmermann, Phil's Pretty Good Software. 1996-01-18
International version - not for use in the USA. Does not use RSAREF.
Current time: 1998/10/30 19:08 GMT

Recipients' public key(s) will be used to encrypt.
A user ID is required to select the recipient's public key.
Enter the recipient's user ID:
**********************************************************************
```

Il PGP sta chiedendo uno o più User ID corrispondenti alle chiavi pubbliche per le quali si desidera crittare il file. Digitiamo la nostra User ID (nell'esempio che stiamo portando avanti: Lametta). In questo modo *solo tu* (o meglio solo chi ha l'accesso alla tua chiave privata) potrai decrittare il messaggio. Comparirà quindi la scritta:

```
***********************************************************
Key for user ID: Lametta <joe@luthorcorp.com>
1024-bit key, key ID C4E35621, created 1998/10/28

Ciphertext file: prova.pgp
*********************************************************
```

Prova.pgp è il file binario (*ciphertext*) in cui è stato crittato il file originale prova.txt. Il PGP assegna automaticamente al file crittato lo stesso nome del file da crittare, modificandone però l'estensione, che nel caso dei file binari è appunto <.pgp>

Nell'esempio appena riportato, non c'è stato bisogno di indicare il path di prova.txt sulla linea di comando del PGP, perché si stava lavorando nella stessa directory c:\pgp\prova dove già si trova il file. Se prova.txt si fosse trovato in una directory diversa, il PGP non sarebbe riuscito a trovarlo e avrebbe dato un messaggio d'errore. Quindi, per crittare un file che non si trova nella stessa directory dove stai lavorando devi indicare il *full path* del file sulla linea di comando del PGP. Se ora cambi directory e digiti:

```
pgp -e c:\pgp\prova\prova.txt
```

vedrai la solita richiesta di battere lo User ID. Una volta fatto, comparirà:

```
*****************************************************************
Key for user ID: Lametta <joe@luthorcorp.com>
1024-bit key, key ID C4E35621, created 1998/10/28

Output file 'c:\pgp\prova\prova.pgp' already exists. Overwrite (y/N)?
*****************************************************************
```

Siccome esiste già un file prova.pgp nella directory c:\pgp\prova, il PGP lo "vede" e chiede se si desidera sovrascriverlo. Battendo y (yes) si otterrà:

```
***************************************
Ciphertext file: c:\pgp\prova\prova.pgp
***************************************
```

Il PGP crea automaticamente il file crittato nella stessa directory in cui si trova il file da crittare. Poiché in questo momento non siamo in tale directory, il PGP si premura di indicare sullo schermo il percorso completo del file prova.pgp[5]. Il file appena creato è un file binario e *non va* inserito nel corpo di un messaggio da postare in rete. Infatti aprendolo con un editor di testo si vedrà una roba del genere:

```
+ "_-+=i-V!__
```

Come abbiamo detto, se questi caratteri venissero inseriti nel corpo di un messaggio, passando attraverso sistemi operativi diversi potrebbero venire alterati, rendendo illeggibile o indecifrabile il messaggio stesso. Per i file che si desidera inserire nel corpo di un messaggio per la posta elettronica si utilizza *sempre* il parametro "-a", il cui compito è creare un *armor* in caratteri ASCII che "protegga" il nostro messaggio da possibili alterazioni. Vediamo un esempio pratico di crittazione di un file in ASCII da spedire nel corpo di un messaggio. Il comando da digitare è:

```
pgp -ea prova.txt
```

Si presenterà la solita schermata con la richiesta di uno o più User ID (ovviamente valgono le stesse osservazioni precedentemente fatte per il path). Alla fine del processo di crittazione il PGP indicherà una schermata un po' diversa:

```
********************************
Transport armor file: prova.asc
********************************
```

Il file creato si chiamerà dunque prova.asc (e non più prova.pgp). L'estensione <.asc> è quella che il PGP assegna ai file crittati in formato ASCII (*transport armor file*)[6].

[5] Esiste il parametro <-o> che consente di creare un file crittato in una directory diversa da quella di default e anche di indicare per il file crittato un *nomefile* diverso da quello di default.

[6] È possibile configurare il file config.sys affinché il PGP crei sempre file crittati in formato ASCII, senza bisogno di digitare il parametro -a ogni volta che si desidera un armor. La scelta di quale settaggio usare di default è squisitamente personale.

Anche i file binari non crittati, come ad esempio quelli compressi con pkzip, per essere inseriti nel corpo di un messaggio devono essere "tradotti" in caratteri ASCII, usando appositi programmi (tipo il noto *uuencode*). È possibile usare anche il PGP per questo scopo di "traduzione in ASCII" senza necessariamente crittare il file. Maggiori informazioni su questo utilizzo si trovano nella documentazione acclusa al programma. Ma torniamo al nostro file prova.asc. Aprendolo con un editor di testo si vedrà un tipico messaggio PGP, che si presenta più o meno così:

```
-----BEGIN PGP MESSAGE-----
Version: 2.6.3i

hIwDwbyr5cTjViEBA/9sqh3uRHUSV0wjkFIp7t36dWRn0vnuZgw4b+15hRNaegeq
ujTHw7SRxR5k7FrR6TJ/S8+BuQ2Fg7yy3P/za07AH3Awdmcu6AR3p0gthNw3mNxC
dMQY/VGovGcu9RGZsJKNRd2GlHgckIIYWr9VyfALfB5ViRWkBXeR/pvD+b31KKYA
AAAn9R17tj2/E9r6JX/N/Z+0jNo4oxzjllp/mmgnaYqfiFZkAD2YAkkh
=9H9y
-----END PGP MESSAGE-----
```

La seconda linea del testo PGP, informa sulla versione del PGP che è stata utilizzata per crittare il messaggio. Le stringhe:

```
-----BEGIN PGP MESSAGE----- e -----END PGP MESSAGE-----
```

ovvero gli *header del PGP*, sono *fondamentali*, in quanto consentono al PGP di riconoscere come crittato il testo tra esse compreso. Bisogna fare attenzione a *non cancellarle né alterarle*, ad esempio quando si vuol fare un "copia e incolla" per importare o esportare un messaggio PGP da un editor di testo al programma che usiamo per la posta elettronica e viceversa, altrimenti il PGP non riconoscerà il testo come crittato, si rifiuterà di decrittarlo e darà solo uno sconsolante messaggio d'errore.

Prima o dopo stringhe -----BEGIN PGP MESSAGE----- e -----END PGP MESSAGE----- può stare senza problemi del testo non-PGP, cioè del semplice testo scritto da noi. Anche se si trova dell'altro testo, il PGP decritterà solo la parte crittata, cioè quella compresa tra le stringhe, fin tanto che (inutile dirlo) le stringhe o il testo fra esse contenuto non sia alterato. Ci sono scopi molto particolari (e attuabili solo da utenti un po' esperti) per cui può essere necessario *togliere gli header del PGP*, cioè cancellare le stringhe di inizio e fine messaggio. Ad esempio in alcune reti telematiche è vietato scambiarsi messaggi crittografati. Per far rispettare un simile divieto i computer che veicolano i messaggi sono dotati di particolari programmi, detti *bot* (da *robot*), che intercettano automaticamente le stringhe -----BEGIN PGP MESSAGE----- e -----END PGP MESSAGE----- e rimandano indietro al mittente il

messaggio che le contiene. Siccome il livello di intelligenza di simili *bot* è ancora sufficientemente basso, un modo per ingannarli è cancellare le stringhe di begin e end. Il destinatario dovrà poi reinserirle a mano (rispettando la posizione originale) e decrittare normalmente il messaggio.

Esistono però (ed esisteranno sempre di più) *bot* meno idioti che riescono a intercettare i messaggi PGP identificando oltre agli *header* anche alcune sequenzialità tipiche degli algoritmi di crittazione usati dal PGP. È questo un classico caso in cui possono risultare utili quei programmi di *steganografia* (trattati più avanti in questo libro) che consentono di nascondere testo, crittato o meno, all'interno di file contenenti immagini, suoni o altro.

Decrittazione di un file

Per decrittare un file crittato, sia binario, sia ASCII, basta battere <pgp *nomefile*> senza altri parametri (valgono sempre le solite considerazioni sul path). Nel nostro caso, per decrittare il file che abbiamo appena creato, bisogna digitare <pgp prova.asc> e si vedrà:

```
************************************************************************
Pretty Good Privacy(tm) 2.6.3i - Public-key encryption for the masses.
(c) 1990-96 Philip Zimmermann, Phil's Pretty Good Software. 1996-01-18
International version - not for use in the USA. Does not use RSAREF.
Current time: 1998/10/30 20:57 GMT

File is encrypted. Secret key is required to read it.
Key for user ID: Lametta <joe@luthorcorp.com>
1024-bit key, key ID C4E35621, created 1998/10/28

You need a pass phrase to unlock your RSA secret key.
Enter pass phrase:
************************************************************************
```

Il PGP riconosce il file come crittato, avvisa che per decrittarlo occorre una chiave privata e chiede la corrispondente passphrase. Dopo averla digitata, se non è sbagliata si vedrà:

```
************************************************************************
Enter pass phrase: Pass phrase is good. Just a moment......
Plaintext filename: prova
************************************************************************
```

Il PGP ha decrittato il file e ha creato un nuovo file a cui ha assegnato automaticamente il nome del file crittato (prova) *ma senza nessuna estensione.* Valgono ovviamente per le decrittazione le stesse considerazioni già fatte per il path. Aprendo con un editor il file c:\pgp\prova\prova si potrà controllare che la decrittazione sia avvenuta senza intoppi.

Sin qui abbiamo parlato di crittare\decrittare un file in binario oppure in ASCII con il *proprio* paio di chiavi, il che è già sufficiente per garantirsi la riservatezza dei dati presenti sull'hard disk. Ma il PGP serve anche e soprattutto per scambiare messaggi con altre persone. In questo caso bisogna ovviamente avere a disposizione la loro chiave pubblica e mettere a loro disposizione la propria. Per fare questo bisogna superare un ulteriore passaggio: *importare le chiavi pubbliche* di altre persone nel proprio pubring.pgp ed *esportare la propria chiave pubblica* in un file che possa essere diffuso via rete o su dischetto.

IMPORTARE ED ESPORTARE UNA CHIAVE

Per ottenere la chiave pubblica di qualcuno si può chiedere direttamente alla persona interessata (che la può consegnare a mano su un dischetto o può spedirla via posta elettronica) oppure (come si vedrà più avanti) si possono utilizzare delle speciali macchine, dette *keyserver*, dedicate alla gestione delle chiavi pubbliche PGP su Internet. Una chiave pubblica ha questo formato:

```
Type Bits/KeyIDDate User ID
pub1024/90653321 1998/02/03 Lex Luthor <lex@luthorcorp.com>

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.3i
Comment: PGP 2.x compatible

mQCNAjTXBdgAAAEEANStJZjQZQpIE/0te0/0o7Y52ESv7usdCrH3BSqLr14QGQpl
+A+0z4UN1MYr+oNS8/xaN1riBdqWtrj80w3vydL12kdHjbtpNT180H1ekU3+1263
/fK1LBXy0E1UWIRExvnVmX9qNd2x0szse+SwTz19PICsJjxEvTt1d0qQZTMhAAUR
tB9MZXggTHV0aG9yIDxsZXhAbHV0aG9yY29ycC5jb20+iQCVAgUQNNcF2Tt1d0qQ
ZTMhAQGUYAP/c+iZSKAZ4juy+cB/H5PdeUSp14NYn8KkAs84tgq+dscG0EQB4vRa
v5Z0kcZTARnV3EU0tEVb+Ube7TSfSjPugD4JEe+/uIRJgJhbfZXbh3uBngLNHshn
NgWBT9/RCBELwvjAv17xu3chhL0xTIq7GNpcy6iJies/6GGndUc2pwg=
=LWqw
-----END PGP PUBLIC KEY BLOCK-----
```

In pratica non è altro che un semplice file di testo in formato ASCII. I *keyfile* sono i file contenenti le chiavi in formato ASCII destinate ad essere

diffuse per posta elettronica. Una chiave pubblica può benissimo essere inserita in un messaggio o in una e-mail. Comunque sia, una volta ottenuta la chiave, va copiata in un keyfile. Supponiamo di aver ricevuto via posta elettronica la chiave pubblica sopra riportata. Per prima cosa la salveremo su un file dal nome luthor.asc (per convenzione si dà al file il nome dello User ID della chiave) in c:\pgp\prova. Per importarla nel proprio keyring si digita (valgono le solite considerazioni sul path):

```
pgp -ka luthor.asc
```

il risultato sarà:

```
*************************************************************************
Pretty Good Privacy(tm) 2.6.3i - Public-key encryption for the masses.
(c) 1990-96 Philip Zimmermann, Phil's Pretty Good Software. 1996-01-18
International version - not for use in the USA. Does not use RSAREF.
Current time: 1998/02/04 00:17 GMT

Looking for new keys...
pub1024/90653321 1998/02/03 Lex Luthor <lex@luthorcorp.com>

Checking signatures...

Keyfile contains:
1 new key(s)

One or more of the new keys are not fully certified.
Do you want to certify any of these keys yourself (y/N)?
*************************************************************************
```

La chiave risulta nuova per il pubring.pgp e *non certificata*: nessuno, cioè, garantisce che quella chiave appartenga effettivamente alla persona a cui si suppone debba appartenere[7]. Il PGP chiede se si desidera certificare personalmente la chiave che si sta introducendo nel pubring.txt. È buona norma non certificare mai una chiave di cui non si è pienamente sicuri, per cui il 99% delle volte a questa domanda si risponde no.

A questo punto il PGP inserirà la chiave pubblica nel pubring.pgp senza dare un ulteriore messaggio di conferma. Si può usare <pgp -kv> per controllare l'avvenuto inserimento. Usato senza ulteriori specificazioni il comando visualizzerà tutte le chiavi, mentre digitando <pgp -kv *User ID*> (in questo

[7] Il meccanismo delle certificazioni (trattato nel precedente capitolo) è piuttosto complesso ma è anche estremamente significativo; tuttavia una sua dettagliata trattazione uscirebbe dai propositi di questa guida introduttiva all'uso del PGP. Chiunque desideri approfondire l'argomento, può sfruttare l'abbondante e dettagliata documentazione che accompagna il pacchetto del software.

caso <pgp -kv luthor>) verrà visualizzata solo la chiave che si desidera[8]. Per inviare ad altre persone la propria chiave pubblica è necessario crearsi un keyfile che la contenga. Il comando per *esportare su file* una public key dal pubring è:

```
pgp -kxa
```

che produce la seguente schermata:

```
***********************************************************************
Pretty Good Privacy(tm) 2.6.3i - Public-key encryption for the masses.
(c) 1990-96 Philip Zimmermann, Phil's Pretty Good Software. 1996-01-18
International version - not for use in the USA. Does not use RSAREF.
Current time: 1998/10/31 00:32 GMT

A user ID is required to select the key you want to extract.
Enter the key's user ID:
***********************************************************************
```

Il PGP chiede lo User ID della chiave da estrarre, in questo caso *il proprio* User ID: Lametta.

```
***********************************************************************
Extracting from key ring: 'c:\pgp\pubring.pgp', User ID "Lametta".

Key for user ID: Lametta <joe@luthorcorp.com>
1024-bit key, key ID C4E35621, created 1998/10/28

Extract the above key into which file?
***********************************************************************
```

e poi chiede in quale file estrarre la chiave. In genere si usa un nome di file che corrisponde allo User ID (Lametta, nel nostro caso) con estensione <.asc>, cioè lametta.asc. Va naturalmente specificato il path se si desidera che sia creato in una directory diversa da quella in cui si sta lavorando, ad esempio:

```
c:\pgp\lametta.asc
```

Il PGP risponderà:

```
*******************************************
Transport armor file: c:\pgp\lametta.asc
Key extracted to file 'c:\pgp\lametta.asc'.
*******************************************
```

[8] La digitazione del comando <pgp -kv> senza ulteriori parametri provoca la visualizzazione di tutte le chiavi contenute nel pubring.pgp, il che può essere un problema quando sono molte. In questo caso si può digitare <pgp -kv | more> per visualizzare una schermata alla volta.

Ora aprendo con un editor di testo il keyfile c:\pgp\lametta.asc si potrà rimirare la propria chiave pubblica. Naturalmente per estrarre la chiave in un'unica operazione, si poteva anche benissimo digitare sin dall'inizio:

```
pgp -kxa lametta c:\pgp\lametta.asc
```

e risolvere tutto in un unico passaggio. Va però ricordato che il PGP è sensibile all'ordine dei parametri sulla linea di comando e finché non si è padroni della sintassi è facile sbagliare. Prova a invertire l'ordine dei parametri e a digitare:

```
pgp -kxa c:\pgp\lametta.asc lametta
```

e vedrai che il PGP interpreta c:\pgp\lametta.asc come lo User ID della chiave pubblica da estrarre: non trovando alcun lametta.asc fra gli User ID l'operazione finisce con il solito, laconico messaggio d'errore.

Crittazione di un messaggio

Siamo ormai a buon punto. Per scrivere un messaggio PGP bisogna innanzitutto preparare un file di testo con il solito editor. Salveremo questo testo in un file dal nome message.txt nella solita directory c:\pgp\prova. Per crittare il messaggio con una o più chiavi pubbliche (in modo che sia decrittabile da una o più persone), si digita il comando:

```
pgp -ea message.txt luthor <tuoUserID>
```

È buona norma crittare ogni messaggio anche con la propria chiave pubblica, in modo da poterlo decrittare in caso di necessità. Per inviare un messaggio crittato, si poteva anche solo battere <pgp -ea message.txt luthor>, ma in questo caso non sarebbe stato più possibile decrittare quel che si era scritto, il che spesso è scomodo. Se si desidera che altre persone siano in grado di decrittare il messaggio, basta battere <pgp -ea prova.txt luthor lametta pippo> - posto naturalmente che si sia importata precedentemente nel proprio pubring.pgp anche la chiave pubblica di Pippo. La risposta del PGP sarà:

```
*************************************************************************
Pretty Good Privacy(tm) 2.6.3i - Public-key encryption for the masses.
(c) 1990-96 Philip Zimmermann, Phil's Pretty Good Software. 1996-01-18
International version - not for use in the USA. Does not use RSAREF.
Current time: 1998/10/31 01:06 GMT
```

```
Recipients' public key(s) will be used to encrypt.
Key for user ID: Lex Luthor <lex@luthorcorp.com>
1024-bit key, key ID 90653321, created 1998/02/03

WARNING: Because this public key is not certified with a trusted
signature, it is not known with high confidence that this public key
actually belongs to: "Lex Luthor <lex@luthorcorp.com>".

Are you sure you want to use this public key (y/N)?
***********************************************************************
```

Come si ricorderà, quando il PGP aveva chiesto se si voleva certificare la chiave pubblica che si stava importando nel pubring, si era risposto di no. Dato che la chiave non è certificata, il PGP ti chiede se vuoi usarla lo stesso. Rispondi di sì (y = yes) e tieni presente che se invece rispondi di no (e per rispondere no *basta battere invio* poiché il no è di default) il PGP non userà la chiave non certificata e critterà il messaggio *solo con la tua chiave* - senza ulteriori avvisi - e quindi il destinatario non potrà decrittare e leggere il messaggio.

Dopo che si è risposto di sì una volta per confermare l'uso di una chiave non certificata, il PGP non chiederà più ulteriori conferme per quella specifica chiave, limitandosi ad avvertirti che non è certificata ogni volta che la si usa. Dopo aver risposto y si vedrà:

```
***********************************************************************
Key for user ID: Lex Luthor <lex@luthorcorp.com>
1024-bit key, key ID 90653321, created 1998/02/03

Key for user ID: Lametta <joe@luthorcorp.com>
1024-bit key, key ID C4E35621, created 1998/10/28

Transport armor file: message.asc
***********************************************************************
```

Il messaggio crittato si trova nel file c:\pgp\prova\message.asc. Si può ora copiarlo e spedirlo con la posta elettronica, ragionevolmente certi che solo il destinatario e noi stessi saremo in grado di decrittarlo e leggerne il contenuto. Per curiosità, ripeti lo stesso comando <pgp -ea message.txt luthor lametta> per vedere come si comporta il PGP usando nuovamente la chiave non certificata.

Una particolarità: dato che il PGP riconosce sulla linea di comando anche una stringa solo parzialmente corrispondente a uno User ID, non è sempre necessario battere lo User ID completo. Ad esempio basta digitare *luthor* oppure *lex* per crittare un messaggio per *lex luthor*. La comodità delle abbre-

viazioni può però a volte creare problemi in caso si posseggano due o più chiavi pubbliche parzialmente coincidenti. Ad esempio se si conserva nel pubring la chiave pubblica di *lex luthor* e quella di *dura lex*, digitando solo il pezzo di User ID *lex*, il PGP fermerà la ricerca della chiave corrispondente al primo User ID comprendente la stringa *lex* che trova nel pubring (cioè critterà di default il messaggio con la chiave pubblica che è stata aggiunta più di recente). In questi casi ci vuole un po' di attenzione nella scelta della porzione di User ID da digitare. Va tenuto presente che il PGP riconosce come parte dello User ID anche l'indirizzo e-mail eventualmente inserito assieme allo User ID. Bisogna fare un po' di prove, digitando porzioni di stringa diverse, per impratichirsi nel meccanismo.

Decrittazione di un messaggio

Ipotizziamo di aver ricevuto un messaggio PGP. Immediatamente lo salviamo su un file in formato ASCII (quasi tutti i programmi di posta elettronica hanno un'opzione per farlo) dandogli il nome che più ci aggrada: in questo esempio receive.txt. Per decrittare il file basta battere il comando PGP seguito dal nome del file:

```
pgp receive.asc
```

la risposta sarà:

```
************************************************************************
Pretty Good Privacy(tm) 2.6.3i - Public-key encryption for the masses.
(c) 1990-96 Philip Zimmermann, Phil's Pretty Good Software. 1996-01-18
International version - not for use in the USA. Does not use RSAREF.

Current time: 1998/10/31 01:43 GMT

File is encrypted. Secret key is required to read it.
Key for user ID: Lametta <joe@luthorcorp.com>
1024-bit key, key ID C4E35621, created 1998/10/28

You need a pass phrase to unlock your RSA secret key.
Enter pass phrase:
************************************************************************
```

bisogna digitare la propria passphrase:

```
**************************************************************************
Enter pass phrase: Pass phrase is good. Just a moment......
Plaintext filename: receive
**************************************************************************
```

A questo punto basta aprire con un editor di testo il file receive (senza estensione perché il PGP salva così di default) e leggere il messaggio. Digitando nuovamente <pgp receive.asc> e battendo invio al momento in cui il PGP chiede la passphrase, si vedrà che nel negare l'accesso alla chiave privata, il PGP riporta il nome degli User ID che sono abilitati a decrittare il messaggio. Questo può essere un modo rapido per controllare se il messaggio è effettivamente crittato con tutte le chiavi che si intendeva usare. Ma può essere anche un momento di debolezza: qualunque altra persona può vedere a chi è destinato quel messaggio, anche se naturalmente non sarà in grado di leggerne il contenuto. Per questo motivo in certe circostanze può essere opportuno inserire come User ID della propria chiave un nome di fantasia o addirittura nessun nome (in questo caso la chiave verrà identificata esclusivamente da una stringa di caratteri alfanumerici generata automaticamente e univocamente per ogni chiave). Per lo stesso motivo si può anche evitare di inserire il proprio indirizzo e-mail.

E A QUESTO PUNTO?

Questo è il momento di usare il PGP operativamente; si può cominciare a distribuire la propria chiave pubblica, magari prima in *matrix* o via e-mail a qualche amico che usa il PGP, per tentare qualche scambio e verificare se tutto è a posto. Se non ci sono problemi, è tempo di diffondere la propria chiave a vasto raggio. Se si desidera, si può anche inserirla su qualche keyserver (le istruzioni sono più avanti). Naturalmente bisogna anche farsi mandare e\o procurarsi le chiavi pubbliche delle persone con cui si corrisponde. Nel frattempo è utile guardarsi bene la documentazione allegata al PGP, approfondendo la padronanza dello strumento, poiché qui si è parlato solo dei primi rudimenti necessari.

Il passo successivo probabilmente sarà l'uso di una *shell*, cioè di un programma di interfaccia, per collegare il PGP direttamente al proprio programma di posta (così da evitare i continui copia/incolla). Prima di concludere il capitolo, vanno ancora posti alcuni problemi piuttosto importanti.

ALCUNE QUESTIONI PRATICHE PER L'USO "SICURO" DEL PGP

La prima cosa da sottolineare è che l'uso di un sistema crittografico come il PGP può generare un falso senso di sicurezza. Bisogna rammentare sempre che PGP di per sé non dice nulla sulla persona con cui si stanno scambiando messaggi (magari critici) e che la regola d'oro rimane sempre la solita: chiunque nel ciberspazio potrebbe essere uno sbirro (e pure fuori). Da sottolineare anche che chiave privata e passphrase del tuo interlocutore potrebbero essere cadute in mano di altri a sua insaputa. Oppure potrebbe anche aver passato dei guai ed essere stato costretto - a tua insaputa - a cedere la sua chiave privata ad altri, che magari stanno continuando a scambiare messaggi con te al posto suo.

Ricorda anche che al momento in cui i guai li passi tu e hai in giro molti messaggi mandati ad altri interlocutori, non puoi essere sicuro che essi non abbiano ceduto sotto pressione la loro chiave, consentendo così la decifrazione dei tuoi messaggi, anche se tu hai tenuto la bocca chiusa... Ma tutto ciò ha poco a che fare con la valutazione di un sistema crittografico e molto a che fare con valutazioni di altra natura.

Diciamo invece che per motivi tuoi hai fiducia nel tuo interlocutore e quindi vuoi essere sicuro che la sua chiave pubblica - sebbene non consegnata a mano, ma pescata da qualche parte o inviata per e-mail - sia davvero la sua. Come fai? Nulla vieterebbe ad un eventuale "terzo incomodo" di inviare una chiave pubblica su un *keyserver* a nome del tuo amico, oppure di intercettare la tua e-mail e di rispondere al posto suo, inviando una chiave a nome del tuo amico e intercettando poi la tua risposta... Il PGP possiede un comando:

```
pgp -kvc <user_ID>
```

che consente di visualizzare il *key fingerprint*, cioè una serie di caratteri alfanumerici che può essere prodotta da una chiave pubblica e *solo da quella*, proprio come un'impronta digitale. Il fingerprint di una chiave pubblica si presenta così:

```
Key fingerprint = 04 FA 8C FC 27 EB 2F 5128 D2 39 84 73 08 30 AA
```

Il fingerprint serve se trovi da qualche parte una chiave pubblica attribuita a Lex Luthor e hai dei dubbi sulla sua veridicità. Se conosci Lex potresti estrarre il fingerprint dalla chiave sospetta, telefonargli e confrontarlo con il suo, in modo da verificare con lui che la chiave sia quella autentica. È un

modo più semplice che non confrontare tutti i caratteri che compongono la chiave pubblica... Non solo. Con il comando:

```
pgp -kvc > fngrprnt.asc
```

si può salvare su file il fingerprint della propria chiave pubblica. Una buona abitudine è quella di inserire il proprio fingerprint nella firma della propria posta elettronica, in modo che compaia su tutti i messaggi che invierai. È così possibile verificare la corrispondenza tra quel fingerprint (diffuso a largo raggio) e quello di tutte le copie della tua chiave pubblica che ci sono in giro[9].

Il PGP offre molte altre possibilità in tema di "sicurezza" che andrebbero approfondite. In primo luogo il sistema della *certificazione delle chiavi*, basato sulla fiducia verso una persona che conosci e con cui è avvenuto uno scambio di chiavi in modo *sicuro*. Questa persona può garantirti, firmandola, una nuova chiave pubblica appartenente a una terza persona che non conosci. Nota però che è preferibile *non usare* il sistema della certificazione piuttosto che usarlo con leggerezza. Considerazioni approfondite su questo aspetto si possono trovare nella documentazione del PGP.

La *firma elettronica* è un ulteriore argomento: è possibile firmare usando la propria chiave privata un messaggio o un testo, che sia crittato oppure in chiaro. Anche qui è bene prestare attenzione: benché attestare in questo modo di essere l'autore di un messaggio possa essere molto utile in certe situazioni, una volta che l'hai firmato *ti sarà molto più difficile negare di esserne l'autore*, qualora se ne presentasse la necessità...

Infine esiste la generazione di un *attestato di firma per un file*, separato dal file stesso di cui si vuole attestare l'autenticità, metodo che viene solitamente usato per file molto grossi o per firmare file eseguibili e programmi. Anche per queste questioni, è necessario leggere approfonditamente la documentazione.

[9] Alcune analisi molto sofisticate hanno smentito la convinzione diffusa ed esposta, per motivi di semplicità, nelle righe precedenti: in realtà il fingerprint non identifica in modo assolutamente univoco una chiave pubblica, in quanto è tecnicamente possibile "pilotare" la creazione di una chiave pubblica affinchè questa presenti un certo fingerprint stabilito a priori (magari proprio quello di una chiave già esistente che si vuole "clonare"). Lo stesso discorso vale per il key ID della chiave. L'unico modo sicuro per identificare una chiave in modo univoco pare pertanto essere quello di verificare *contemporaneamente* fingerprint, key ID e dimensione della chiave (in quanto nella creazione di una chiave con fingerprint "pilotato" si perde il controllo sulla sua dimensione in bit). Si tratta di una delle questioni che rappresentano probabilmente lo stato dell'arte nella ricerca di punti deboli nel PGP.

APPENDICE: I KEYSERVER

I *keyserver* sono particolari server presenti su Internet, dedicati al deposito e al prelievo delle chiavi pubbliche. Sono in rete tra loro, per cui ogni chiave immessa in un server viene diffusa anche sugli altri. Per ricevere o immettere chiavi bisogna inviare un'e-mail all'indirizzo del keyserver. In Italia è solitamente utilizzato:

```
pgp-public-keys@dsi.unimi.it
```

I comandi al keyserver vanno scritti nel "subject" dell'e-mail: per *ricevere* una chiave pubblica basta inviare al keyserver una e-mail, contenente nel subject il comando GET seguito dallo User ID della chiave richiesta. Chi vuole la chiave pubblica di Lex Luthor scriverà un e-mail così:

```
To: pgp-public-keys@dsi.unimi.it
From: joe@luthorcorp.com
Subject: GET lex@luthorcorp.com
```

o anche solo GET luthor, senza aggiungere nulla nel corpo dell'e-mail.

Per *depositare* la propria chiave occorre spedire un'e-mail al keyserver, con il comando ADD nel subject. Nel corpo del messaggio va copiata la propria chiave, estratta col comando <pgp -kxa> in modo da essere, come abbiamo visto, in formato ASCII. Ecco un'e-mail con cui si chiede di depositare una chiave:

```
To: pgp-public-keys@dsi.unimi.it
From: lex@luthorcorp.com
Subject: ADD
________________________________

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.3i
Comment: PGP 2.x compatible

mQCNAjTXBdgAAAEEANStJZjQZQpIE/0teO/Oo7Y52ESv7usdCrH3BSqLr14QGQpl
+A+0z4UNlMYr+oNS8/xaNlriBdqWtrj8Ow3vydL12kdHjbtpNT180H1ekU3+1263
/fKlLBXyOE1UWIRExvnVmX9qNd2xOszse+SwTzl9PICsJjxEvTt1dOqQZTMhAAUR
tB9MZXggTHV0aG9yIDxsZXhAbHV0aG9yY29ycC5jb20+iQCVAgUQNNcF2Tt1dOqQ
ZTMhAQGUYAP/c+iZSKAZ4juy+cB/H5PdeUSp14NYn8KkAs84tgq+dscG0EQB4vRa
v5ZOkcZTARnV3EU0tEVb+Ube7TSfSjPugD4JEe+/uIRJgJhbfZXbh3uBngLNHshn
NgWBT9/RCBELwvjAv17xu3chhL0xTIq7GNpcy6iJies/6GGndUc2pwg=
=LWqw
-----END PGP PUBLIC KEY BLOCK-----
```

Per altre informazioni sulle funzioni di un keyserver, si può inviare allo stesso un'e-mail con scritto HELP nel subject della stessa.

Visto? Non era poi così difficile da trovare il fottuto PGP. E nemmeno da usare, quando ci avevi messo un po' di tempo per capire come si fa.

Tempo tre ore e Luthor e io eravamo già in contatto. Niente male per uno che ha mollato la scuola a dieci anni, eh? Ma il diploma in furto con scasso, quello almeno avrebbero potuto darmelo. Già... ma questa bottiglia ormai è quasi vuota amico. Facciamone portare un'altra. Cosa stavo dicendo? Ah sì. Che una volta capito il trucchetto, era facile. Anche perché noi problemi di sapere se davvero stavamo parlando uno con l'altro non ne avevamo mica. Nella valigetta c'era un dischetto con le mie chiavi e con la chiave pubblica di Lex. Un'altra copia ce l'aveva solo lui, e bastava "importarle nei mazzi di chiavi"... Sin troppo facile, da' retta a me. Perché il vecchio Luthor era un po' in ansia, e dopo il primo messaggio è stato un fiume di messaggi da far sembrare un rigagnolo il Mississipi quand'è in piena. È successo questo, è successo quest'altro. Fai quello, fai quell'altro. Sei sicuro di questo, ricordati di quello. Bla bla bla bla bla e bla. Eh, un po' lo capivo Lex. Il colpo più importante della sua vita, e lui era lontano mille miglia. Ottimo per pararsi il culo, ma per uno come lui, abituato a far scattare tutti con uno schiocco di dita, starsene via dall'azione e aspettare i miei rapporti era proprio un bel rodimento di culo. Quando se ne lamentava mi faceva proprio tanta compassione. Come un alligatore con le emorroidi. Bah. Comunque amico, certe cose tornava comodo saperle anche a me. Secondo le spie di Luthor il nostro amato Sindaco ormai aveva sotto gli occhi due borse tanto grosse da poterci far entrare una Cadillac e relativo rimorchio, mi spiego? E aveva mandato a chiamare Superman dopo due giorni di SuperRicerche a vuoto. E gli aveva detto senza giri di parole davanti a tutti i Pezzi Grossi di Metropolis che era l'ora di ripagare una buona volta tutte quelle montagne di bistecche con patate e torta di mele che si era pappato per anni a spese della Città. E che se non riusciva a trovare la Bomba, lui come sindaco era fottuto, e il massimo che poteva sperare dopo era un bel posto di viceaddetto alla manutenzione dei cessi comunali. Ma se lui finiva a fare il grattamerda, Superman sarebbe finito assieme a lui a fare la SuperCisterna dei pozzi neri. E si sarebbe pure sentito fortunato a farsi 4 dollari e 75 al giorno aspirando la merda col SuperRisucchio,

perché forse se ne sarebbe ricoperto abbastanza da non farsi più riconoscere e additare da tutti come il SuperCoglione che era. Heh... Il nostro Omino d'Acciaio è abituato a essere coccolato come il più bravo boyscout di Metropolis, e ad essere preso a pacche sulle spalle da tutti , mica ad essere trattato come quel povero demente che è. Ha irrigidito il mascellone di granito, ha risposto secco secco "Sissignore" ed è ripartito in volo verticale come un missile dopo aver fatto il saluto militare. Lasciandosi dietro una serie di buchi nei soffitti del Municipio sino al tetto, tanto che il Sindaco si è chiesto se non aveva calcato troppo la mano, mi spiego? A parte queste notizie, Luthor insisteva un casino che dovevo continuare a studiare le istruzioni sui sistemi per tenerci in contatto, perché per ora tutto filava liscio, ma le cose potevano mettersi male in fretta. E allora, dato che non avevo nemmeno granché da fare per il momento, mi sono rimesso a leggere....

File System Crittati

di Marta McKenzie

Finora sono stati considerati esempi di crittografia applicati a singoli file. In realtà qualsiasi informazione contenuta in un computer è riducibile a gruppi di bit e quindi queste tecniche sono applicabili anche a corpi di informazione più estesi e complessi di file singoli.

In particolare, qualunque utente del PGP si rende rapidamente conto di quanto possa essere utile avere a disposizione un sistema per mantenere riservato il contenuto di intere cartelle o directory o addirittura di intere porzioni del proprio disco fisso o di altre memorie di massa. Questo breve capitolo si occuperà della possibilità di applicare tecniche di crittografia non più su file presi singolarmente bensì su generici insiemi strutturati di informazioni.

A tali insiemi di informazioni ci si riferisce a volte con il concetto di *file system*. Un file system è costituito dall'intera struttura utilizzata dal computer per accogliere le informazioni sulla memoria di massa: un file system comprende quindi i file, ma anche i vari livelli di directory o cartelle che li contengono e le particolari aree del disco fisso che accolgono le informazioni relative a tali cartelle e al metodo con cui sono registrate sul disco (FAT o equivalenti, boot sector, ecc.).

Esistono sistemi attraverso i quali è possibile applicare crittografia e steganografia a interi file system: questo significa che, senza le opportune chiavi di accesso, un intero disco fisso può risultare illeggibile o addirittura scomparire (è questo, in breve, l'obiettivo della steganografia, tecnica a cui si accennerà più avanti e che verrà trattata in modo particolareggiato in uno dei prossimi capitoli; i più curiosi possono anche andarsi a leggere subito il capitolo sulla steganografia e poi proseguire la lettura di questo capitolo). Tali

sistemi sono perlopiù ancora in fase di sperimentazione e le loro caratteristiche stanno cambiando abbastanza in fretta. Ci sono molte idee che ci vengono in mente pensando a cosa potrebbe essere utile in un campo come questo, ma da quanto ci risulta buona parte di queste idee non sono ancora state implementate da nessun software.

Pensiamo a cosa potrebbe essere utile in caso, ad esempio, di perquisizione da parte del nemico: una porzione di disco fisso, completa di cartelle e file, che risulta invisibile a occhi estranei. Attraverso tecniche steganografiche, la stessa esistenza della partizione non è tecnicamente dimostrabile senza essere in possesso delle chiavi di accesso. Senza poter dimostrare l'esistenza della partizione, non è ovviamente nemmeno più possibile chiedere al proprietario di "confessare" la chiave per accedervi. Non solo: all'immissione della passphrase "A" la partizione diventa visibile, ma all'immissione di una particolare passphrase "B" la partizione viene irrimediabilmente distrutta e con essa i dati che contiene. Se anche si decide di "confessare" la chiave, è possibile rendersi conto che si tratta della passphrase "B" solo *dopo* che essa ha già svolto il suo compito.

Queste caratteristiche e altre ancora non fanno parte, per ora, di alcun sistema "ufficiale" disponibile in rete. In alcuni casi sono già disponibili, però, possibilità molto sofisticate, come i file system steganografati sotto il sistema operativo *Linux* di cui si parlerà fra poco. È importante rendersi conto che simili caratteristiche tecniche sono ormai ampiamente possibili alla luce del bagaglio di conoscenze esposto nei capitoli precedenti. Chiunque disponga di sufficienti competenze tecniche è in grado già ora di costruirsi il proprio personalissimo campo minato digitale ed eventualmente metterlo a disposizione degli altri come è già successo per molto software. Come al solito, la stessa conoscenza di queste possibilità da parte degli organi di controllo e repressione sociale non rende loro la vita più semplice: di fronte alla possibilità di rendersi invisibili, una legge che intenda punire ciò che per definizione non si vede è destinata solamente a coprirsi di ridicolo.

Il problema del sistema operativo: in ambiente DOS

Purtroppo, un aspetto negativo dei sistemi in grado di gestire file system crittati è quello che, al di là degli aspetti generali, tali sistemi sono necessariamente legati al sistema operativo utilizzato sul computer. Questo significa

che esistono sistemi in grado di lavorare sotto MS-DOS, sotto Windows 3.1, sotto Windows 95/98 o NT, sotto Linux e via di questo passo. Ognuno di questi sistemi avrà caratteristiche proprie. Non sarà quindi possibile proseguire la discussione senza fare riferimento a un software o almeno a un sistema operativo ben preciso.

A causa della sua popolarità, è sotto MS-DOS che probabilmente sono disponibili i sistemi per file system crittati più affidabili e conosciuti. Uno di questi è *Secure Drive*, programma freeware distribuito nell'ambito del progetto GNU e di cui sono quindi disponibili i sorgenti. Come per qualsiasi sistema di crittografia, la disponibilità dei codici sorgenti (che permettono di avere una conoscenza perfetta dell'effettivo funzionamento del programma) costituisce una caratteristica necessaria (ma non sempre sufficiente) affinché il sistema possa essere considerato sufficientemente sicuro.

Secure Drive richiede una partizione apposita sul disco fisso. Tale partizione e tutto il suo contenuto vengono sottoposti a crittazione con l'algoritmo IDEA, attualmente uno degli algoritmi di crittografia più robusti tra quelli conosciuti (è l'algoritmo che viene usato, assieme all'RSA, dal PGP). La chiave di accesso alla partizione può essere costituita da un piccolo file segreto, da una passphrase o da entrambi (analogamente al PGP). Come viene suggerito dalla documentazione allegata al programma, questa possibilità può essere sfruttata in modo intelligente in determinate occasioni: se per l'accesso è necessario un file (chiamato *keyfile*) e questo keyfile risiede esclusivamente su un dischetto che viene fisicamente distrutto, l'accesso alla partizione diventa tecnicamente impossibile. Questo fatto può essere confermato da una perizia condotta da consulenti esperti e può risultare molto utile nel caso il nemico "ordini" la consegna delle chiavi di accesso.

Inoltre, è anche possibile configurare il programma in modo che visualizzi un messaggio di errore relativo al "keyfile distrutto", pur consentendo comunque l'accesso con la sola passphrase. Il nemico sarà indotto a credere che l'accesso non sia più tecnicamente possibile. Se anche sorge il dubbio che le cose non stiano così, l'unico modo per dimostrarlo e quindi chiedere legittimamente di "favorire" la passphrase, è proprio quello di riuscire prima ad accedere alla partizione utilizzando la passphrase stessa.

Sulla base di *Secure Drive* è stato ricavato un altro programma, chiamato *Secure Device*, anch'esso distribuito completo di sorgenti e destinato al sistema operativo MS-DOS, che possiede alcune ulteriori caratteristiche. La più interessante è costituita dal fatto di non richiedere una partizione apposita per poter funzionare. La creazione di una partizione (cioè di un ulteriore disco fisso "logico" all'interno di un disco fisso fisico) è spesso un'operazio-

ne rischiosa e delicata quando sul disco sono già ospitati dei dati, e *Secure Device* evita di dover compiere questa operazione.

Un ulteriore programma per MS-DOS (gratuito per l'uso privato) è *Secure File System*, sviluppato con l'aiuto di alcuni collaboratori di Phil Zimmerman (l'autore del PGP) e dunque quantomeno degno di attenzione. Anche questo programma, pur richiedendo una partizione apposita, offre numerose funzioni utili. Tra queste, la possibilità di "smontare" (cioè rendere non più accessibile) velocemente il disco crittato in caso di emergenze: una volta decrittati i dati, infatti, per riportarli istantaneamente alla loro forma crittata è sufficiente premere una particolare combinazione di tasti configurata in fase di installazione. È anche possibile prevedere che un disco si "smonti" da solo automaticamente se non viene utilizzato oltre un certo periodo di tempo. Entrambe queste funzioni, come si può capire, sono utili nel caso in cui il nemico compia un'irruzione improvvisa nella speranza di "catturare" un sistema informatico già avviato in cui siano già state immesse le chiavi di accesso.

In ambiente Windows 95 e NT

I programmi citati finora sono stati concepiti per l'utilizzo sotto MS-DOS. Sono sempre più numerosi, tuttavia, coloro che utilizzano il computer usufruendo della semplicità e comodità di Windows. In questo caso sorgono però alcuni problemi: sebbene sia ancora possibile, con alcune limitazioni, usare questi programmi con le vecchie versioni di Windows (Windows 3.1 e 3.11), sotto Windows 95/98 la sicurezza e stabilità complessiva del sistema vengono compromesse.

In questo momento pare non esistano programmi completamente affidabili e gratuiti per gestire file system crittati sotto Windows 95/98 o NT. Ci si limiterà quindi ad accennare brevemente a due programmi che, benché non offrano la sicurezza garantita dai precedenti e non siano del tutto gratuiti, possono comunque costituire delle ragionevoli alternative per chi intenda in ogni caso sperimentare la gestione di dati crittati sotto le nuove versioni di Windows.

Il primo programma è *BestCrypt NP*, prodotto dalla Jetico, Inc. Studiato appositamente per Windows 95/98, offre numerose funzioni utili, comprese le opzioni di "smontaggio" rapido automatico dei dati attraverso una combinazione di tasti o un intervallo di timeout. Tra gli algoritmi di crittografia

supportati, si può scegliere, accanto al vecchio e ormai insicuro DES, anche il *Blowfish*, che si sta affermando come uno degli algoritmi più veloci e sicuri, ottima alternativa all'IDEA. È inclusa nel pacchetto software anche un'utility per cancellare i file in modo sicuro, operazione di cui si dirà tra poco. L'uso di *BestCrypt NP* è gratuito per i primi 30 giorni di utilizzo, oltre i quali il programma accede ai dati in sola lettura e secondo gli autori diventa necessario acquistare la licenza d'uso che costa circa 90 dollari.

Il secondo programma cui vale la pena di accennare è *Kremlin*, che a differenza dei precedenti non si presenta come un sistema specificamente studiato per la gestione di file system crittati, bensì come insieme di programmi per la tutela della privacy; tra le varie funzioni sono infatti comprese: cancellazione sicura dei file, cancellazione sicura dello swap file di Windows, gestione di messaggi di posta elettronica, esecuzione di operazioni avviate automaticamente alla partenza o allo spegnimento del computer, crittazione di file singoli e di intere directory. Quest'ultima funzione è quella che più si avvicina alla categoria di programmi presentati in questo capitolo. Non si tratta, tuttavia, di una reale gestione di file system veri e propri: ad essere crittate sono semplicemente le directory che indichiamo, e non intere unità logiche. Ma la cosa più importante è che la crittazione, per quanto utilizzi algoritmi sicuri e conosciuti, avviene in modo fondamentalmente diverso rispetto ai casi precedenti.

In un sistema *sicuro* di gestione di file system crittati, infatti, tutti i dati vengono crittati e decrittati in tempo reale, mano a mano che l'utente li richiede. Se si ha bisogno di accedere a un determinato file, ad esempio, il sistema lo renderà disponibile in chiaro solo per il tempo in cui il file viene effettivamente mantenuto aperto (ad esempio per la modifica con un word processor, nel caso di un documento scritto), crittandolo nuovamente non appena il file viene chiuso. Tutto questo avviene in modo completamente trasparente per l'utente, che si comporta come se il file system fosse effettivamente disponibile decrittato sotto i suoi occhi; in realtà i dati in chiaro mantenuti in memoria sono strettamente limitati a quelli che si stanno effettivamente utilizzando, e l'intero file system non viene comunque mai conservato in chiaro sulla memoria di massa. Questo serve a proteggere il sistema nell'eventualità che esso cada improvvisamente in mano al nemico: è solo con questa tecnica, infatti, che è possibile prevedere opzioni di "smontaggio" rapido come quelle che abbiamo visto offrire dai programmi precedenti. *Kremlin*, al contrario provvede a una completa decrittazione dei file non appena si immette la passphrase, e i file decrittati vengono mantenuti in chiaro sul disco fisso fino a quando non si immette nuovamente una passphrase per una nuova crittazione, operazione che richiede diversi minuti se i dati hanno dimensioni superiori a qualche megabyte.

Esiste inoltre un'ulteriore situazione nella quale la crittazione trasparente e in tempo reale può prevenire conseguenze disastrose. Ad esempio, si è già notato in casi concreti che, quando gli organi di polizia progettano una perquisizione avente come specifico obiettivo apparecchiature informatiche e dati digitali, vengono talvolta osservate precauzioni particolari: una di queste è il distacco della corrente elettrica dall'appartamento soggetto a perquisizione, appena prima dell'irruzione (presumibilmente per impedire ai sospettati di cancellare prove o indizi). Gli effetti di un simile espediente sono totalmente efficaci se si utilizza *Kremlin*, piuttosto che un prodotto come *BestCrypt NP*. Nel primo caso infatti, se i dati erano stati decrittati dall'utente nel momento in cui è stata interrotta l'alimentazione del computer, essi rimangono liberamente disponibili in chiaro sul disco fisso, pronti per essere letti non appena il computer verrà nuovamente acceso. Al contrario, se si utilizza *BestCrypt NP* o uno qualsiasi degli altri programmi presentati, il distacco della corrente cancellerà semplicemente dalla memoria volatile (memoria RAM) il contenuto in chiaro dei file aperti in quel preciso istante, lasciando sul disco solamente la versione crittata dell'intero file system. Visto che siamo in argomento, ricordiamo che esistono due modi efficaci per proteggersi dalle interruzioni di corrente, casuali o volute che siano: il primo è quello di dotarsi di un gruppo di continuità, apparecchiatura moderatamente costosa che funge da piccolo accumulatore; il secondo è quello di utilizzare personal computer portatili, che possono continuare a funzionare con la propria batteria anche se privati improvvisamente della corrente di rete.

In ogni caso, *Kremlin* è disponibile in versione dimostrativa gratuita. Per utilizzare la versione completa, che consente l'accesso agli algoritmi di crittografia più sicuri (come IDEA e *Blowfish*), secondo gli autori è necessario acquistare la licenza venduta a 35 dollari.

È da tenere presente che sia *Kremlin*, sia *BestCrypt NP* come tutti i programmi commerciali vengono distribuiti senza la possibilità di esaminare i sorgenti, impedendo quindi di verificare in prima persona l'effettivo funzionamento interno dei programmi.

Linux: un'opportunità in più

Per gli utenti del sistema operativo *Linux*, esiste non solo la possibilità di applicare la crittografia a un intero file system, con modalità del tutto analoghe

a quelle già discusse, ma anche la possibilità di steganografare il file system stesso dentro immagini o suoni, usando una semplice tecnica iniettiva come quella che verrà presentata nel capitolo sulla steganografia. Il risultato è quello di avere un disco fisso "logico" (utilizzabile come un normale disco) nascosto all'interno di suoni o immagini, in modo tale che questi suoni o immagini continuino a essere uditi o visualizzati normalmente senza destare sospetti.

Per poter utilizzare queste funzioni sul proprio sistema è necessario un minimo di competenza tecnica (che d'altra parte è indispensabile anche solo per utilizzare *Linux* come principale sistema operativo). Occorre infatti ricompilare il kernel (la parte essenziale del sistema operativo), dopo avere applicato ai sorgenti le opportune modifiche (patch) reperibili al seguente URL: ftp://csclub.uwaterloo.ca/pub/linux-stego/index.html.

Una volta modificato il proprio kernel, l'uso congiunto delle funzioni crittografiche e steganografiche è senza dubbio il più interessante tra gli esempi proposti.

In pratica si procede nel seguente modo. Si sceglie un file contenitore abbastanza ampio (per il significato di "file contenitore" e per una discussione generale sulla steganografia, si veda il relativo capitolo), per esempio un file audio ottenuto digitalizzando un messaggio vocale. Mediante gli opportuni comandi si istruisce il sistema a trattare i bit meno significativi (e solo quelli) del file audio come se fossero un normale dispositivo di memorizzazione. A questo punto è possibile formattare questo spazio virtuale proprio come se fosse un comunissimo disco fisso o floppy disk. A formattazione ultimata, quello che si ottiene è un vero e proprio file system del tutto indistinguibile da quelli ottenuti con metodi più tradizionali. È quindi possibile accedervi in lettura e scrittura, creare o rimuovere directory e file, ecc.

Se il file system è stato anche crittato (come in effetti dovrebbe essere, per rispettare le condizioni di sicurezza), al momento in cui si chiede di utilizzarlo è necessario fornire anche la password opportuna.

Quando si è finito di utilizzare il file system è necessario smontarlo, dopodiché di esso non rimane più alcuna traccia. Se al file system sono state apportate delle modifiche (ad esempio è stato creato o cancellato un file) il file contenitore risulta naturalmente a sua volta modificato, ma solo nei bit meno significativi. Questo significa che il file audio continua a essere perfettamente ascoltabile, senza che si possa udire una significativa differenza rispetto alla versione originale.

Si tenga presente che, nel caso dei file audio in formato .au la capienza del file system è un ottavo della dimensione del file (di fatto è un po' meno, se si

considera lo spazio che il sistema operativo riserva per le informazioni atte a gestire lo stesso file system), il che vuol dire che per avere l'equivalente di un floppy disk ad alta densità abbiamo bisogno di un file audio di circa 12Mb.

Si notino infine le particolari garanzie offerte dall'accoppiata crittografia + steganografia: senza disporre della password, è impossibile non solo leggere il contenuto di file e directory, ma anche semplicemente dire con certezza se un file audio (o un'immagine) contenga o meno al suo interno un file system.

MILLE TRACCE DA RIMUOVERE

A conclusione del capitolo facciamo una breve digressione su una questione non legata direttamente alla crittazione dei file system, ma che è comunque di primaria importanza nella gestione generale di file e dati sensibili su supporti digitali.

Ipotizziamo una normale sessione di lavoro al computer: si naviga in rete alla ricerca di informazioni, con un word processor si creano alcuni nuovi documenti, alcuni file vengono spostati da una cartella a un'altra, altri ancora vengono cancellati. Queste operazioni lasciano più tracce di quante ne immaginiamo. I documenti visualizzati con un browser web come Netscape o Explorer, ad esempio, rimangono in una particolare zona del disco (chiamata *cache*) anche dopo la disconnessione dalla rete e permettono a chiunque metta le mani sul nostro computer di ricostruire le nostre ultime navigazioni on-line. Quasi tutti i nuovi programmi applicativi (word processor, database, eccetera) visualizzano, per poterli caricare più velocemente, i nomi degli ultimi file aperti nelle sessioni di lavoro precedenti. Lo stesso Windows 95/98, nel menu Avvio, presenta una voce relativa ai "Dati recenti" che offre una panoramica immediata sulle ultime attività dell'utente. In particolari casi queste informazioni possono contenere anche dati riservati che si vogliono mantenere al riparo da occhi indiscreti. Le soluzioni sono abbastanza semplici, ma piuttosto scomode da applicare regolarmente. Possono venirci in aiuto programmi specializzati che, come il già citato *Kremlin*, provvedono a ripulire automaticamente, a ogni spegnimento del computer, le informazioni relative alla cache del browser o al menu "Dati recenti" di Windows.

Più complicato e potenzialmente molto più pericoloso è il problema della cancellazione dei file. Le normali operazioni richieste per cancellare un file (ad esempio il comando *"del"* sotto DOS o il trascinamento nel cestino

sotto Windows) in realtà *non cancellano affatto* il contenuto del file vero e proprio: si limitano a cancellare le intestazioni del file dall'elenco interno utilizzato dal file system. In termini pratici, si segnala al computer che lo spazio su disco occupato da quel determinato file non contiene più informazioni utili e può quindi essere utilizzato da altri file non appena si presenti il bisogno di scrivere nuovi dati su disco. Di fatto, però, il vecchio contenuto rimane scritto su disco fino a quando non viene sostituito da nuovi dati che ne vanno a occupare l'esatta posizione. È piuttosto difficile prevedere quando potrà avvenire la sovrascrittura del vecchio contenuto da parte di nuove informazioni (perché le nuove informazioni possono essere scritte in qualsiasi settore libero del disco). Il risultato è che i file "cancellati" possono in realtà continuare a essere disponibili sul computer per un periodo di tempo imprecisato (giorni, mesi o anni a seconda di una molteplicità di fattori).

Tutto questo è ormai ampiamente conosciuto tra gli utenti di Windows, che hanno a disposizione un comodo "cestino" dal quale recuperare i file cancellati per errore. Era invece sbalorditivo, per alcuni dei più ingenui utenti DOS, scoprire che appositi programmi potevano far "resuscitare" informazioni che credevano cancellate per sempre. Naturalmente, a volte si trattava di una sorpresa piacevole (quando i file recuperati erano stati cancellati per errore), altre volte meno (quando i file erano stati cancellati volutamente, perché contenenti informazioni riservate). Anche per chi utilizza Windows, comunque, è importante ricordare che la stessa eliminazione dal cestino *non* costituisce una cancellazione sicura.

La portata di questo problema è evidente: è perfettamente inutile dotarsi anche del migliore sistema per la crittazione di file system, se non ci si rende conto che i file cancellati durante le normali operazioni di lavoro su file system tradizionali rimangono disponibili sul disco.

Per fortuna, una volta chiarito il problema, le soluzioni sono semplici e abbastanza sicure. È sufficiente dotarsi di una qualsiasi delle numerose utility studiate appositamente per la cancellazione sicura dei file. Il principio di funzionamento di questi programmi, in generale, è molto semplice: anziché informare il sistema operativo che lo spazio occupato dal file *luthor.txt* non contiene più informazioni utili e può quindi essere rimpiazzato da qualcos'altro alla prima occasione, il software per la cancellazione sicura provvede *prima* a rimpiazzare il contenuto del file sovrascrivendolo con informazioni casuali, e *poi* a dichiarare quello spazio disponibile per il sistema operativo. Il risultato è che l'operazione è leggermente più lenta (è necessario scrivere subito su disco una quantità di byte corrispondente a quelli da cancellare), ma la cancellazione è definitiva.

Si lascia a chi legge il compito di trovare in rete i programmi per la cancellazione sicura più adatti alle proprie esigenze, ma è utile ricordare che lo

stesso PGP offre un servizio di questo tipo (con l'opzione -w descritta nella documentazione ufficiale del PGP).

La cancellazione sicura è un problema che riguarda anche lo swap file creato dai sistemi operativi che fanno uso della memoria virtuale su disco (come ad esempio Windows). Come si è già accennato nel capitolo sulla crittografia, lo swap file può contenere informazioni delicate (comprese le passphrase dei nostri sistemi crittografici) e dovrebbe quindi subire lo stesso processo di cancellazione sicura destinato a ogni file "sensibile". Essendo lo swap file un file piuttosto particolare (sia per dimensioni sia per altre caratteristiche tecniche) ed essendo riscritto dinamicamente durante le sessioni di lavoro, per la sua cancellazione sistematica è consigliabile utilizzare programmi specifici (il già citato *Kremlin*, o il più semplice *Scorch*).

Concludiamo con un po' di paranoia: come d'uso nei circoli criptoanarchici e cypherpunk, sono state fatte notevoli speculazioni teoriche sulle possibilità di recuperare un file anche dopo essere stato sovrascritto con gli appositi programmi. Ipotizzando che il nemico abbia in mano strumentazioni tecniche particolari e una fortissima determinazione, sono emerse due possibili linee di attacco per il recupero di un file sovrascritto. Si tratta, è bene ribadirlo, di possibilità del tutto teoriche e particolarmente paranoiche, che nessuno è stato ancora in grado di attuare concretamente, nemmeno in via sperimentale. Entrambe sfruttano particolari accorgimenti tecnici che in questa sede è possibile spiegare solo semplificando molto le cose.

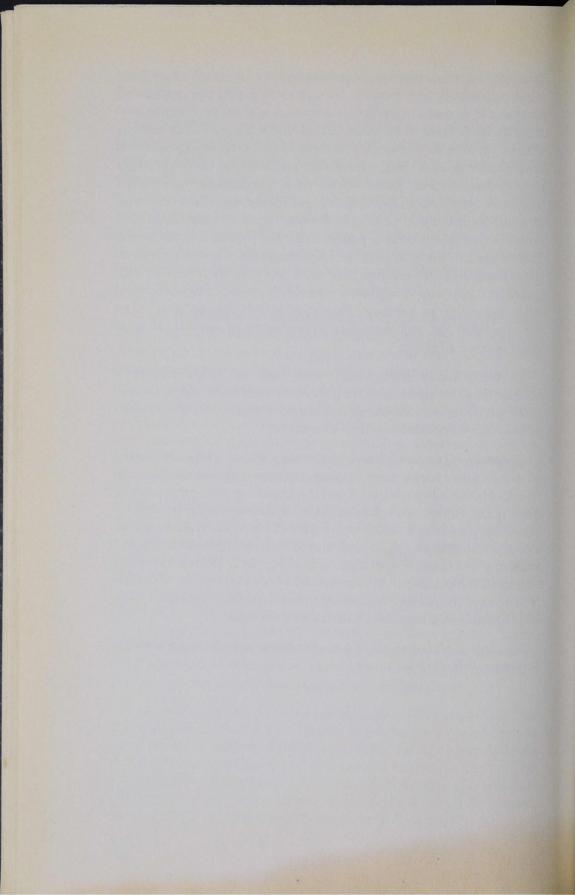
La prima possibilità è data dal fatto che il disco fisso è suddiviso in elementi discreti, chiamati tracce (tracks); le testine del disco sono vincolate a scrivere seguendo queste tracce. Con l'uso, e a seconda della qualità dell'hardware, l'allineamento tra tracce e testine può variare leggermente. Quando si scrivono su disco nuove informazioni lo scarto tra l'allineamento attuale e quello precedente può far sì che, ai margini delle tracce vengano conservati piccolissimi residui di scritture precedenti. Attraverso la scansione del disco con un microscopio elettronico è teoricamente possibile leggere questi residui e tentare di reinterpretare i file che rappresentavano. Si tratterebbe comunque di informazioni limitatissime, che solo con grandissima fortuna (o sfortuna) potrebbero rappresentare dati importanti.

La seconda possibilità si affida ai residui magnetici del disco. Sui supporti magnetici come i dischi fissi e i floppy, i singoli bit 1 e 0 (ricordiamo che tutta l'informazione digitale è riducibile, in ultima analisi, a sequenze binarie di 1 e 0) vengono registrati attraverso polarità magnetiche opposte. Ma mentre le unità di informazione 1 e 0 sono entità discrete, le registrazioni magnetiche sono quantità continue. Questo significa che un determinato bit 1 viene registrato magnetizzando una parte del disco con un valore che si

"avvicina" significativamente alla polarità prevista per i bit di valore 1, ma senza raggiungerla perfettamente. Lo scarto sarà in ogni caso minimo e il computer lo ignorerà, interpretando comunque quel valore di magnetismo come un bit di valore 1. Lo stesso vale, naturalmente, per i bit di valore 0. Quando un bit viene sovrascritto (ad esempio da uno dei programmi per la cancellazione sicura dei file), si aprono due possibilità: nel caso un bit venga sovrascritto da un altro bit dello stesso valore il magnetismo verrà confermato, cioè spinto ancora di più verso la polarità attuale; nel caso invece un bit venga sovrascritto da un altro bit di valore opposto il magnetismo verrà invertito, cioè spinto verso la polarità magnetica opposta a quella attuale. Questo significa che quanto più un campo magnetico si avvicina con precisione a una delle due polarità, tanto più è probabile che, "nei suoi stadi precedenti" quel campo magnetico ospitasse lo stesso bit; viceversa, tanto più il magnetismo sia impreciso, allontanandosi dalla polarità "pura", tanto più è probabile che quel posto fosse in precedenza occupato da un bit di valore opposto all'attuale.

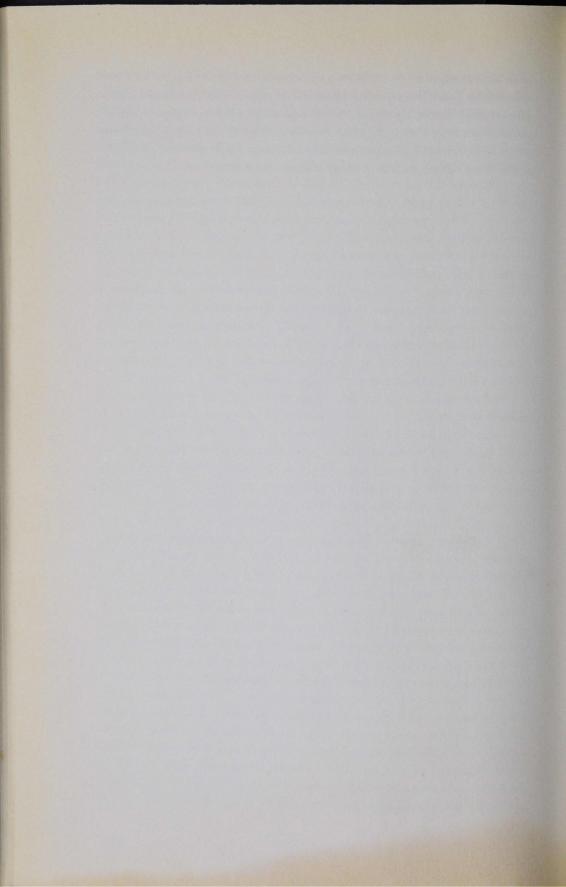
Queste differenze di magnetismo sono totalmente insignificanti per il normale hardware informatico, ma possono in teoria essere rilevate da strumenti specifici e, una volta rilevate, analizzate per tentare di recuperare informazioni utili alla ricostruzione di files sovrascritti.

Entrambe queste possibilità rappresentano il massimo della paranoia che si è riusciti finora a immaginare su questo argomento. Per quanto rappresentino dei pericoli solo teorici, quasi tutti i programmi specifici per la cancellazione sicura offrono una soluzione molto semplice anche nei confronti di queste due possibilità: è sufficiente che il file da cancellare venga immediatamente sovrascritto *più volte*, anziché una sola volta, per minimizzare le informazioni potenzialmente contenute negli scarti tracce/testine e nei residui magnetici. Il meccanismo di funzionamento è identico e l'unica differenza per l'utente è che per ogni cancellazione dovrà aspettare un periodo di tempo proporzionale al numero di sovrascritture da eseguire.

Ognuno di noi saprà valutare se quei trenta secondi in più sono più importanti delle proprie paranoie.

E ho fatto bene a sciropparmi ancora un po' di quella roba. Uno pensa di cancellare le cose che ha scritto e invece gli restano lì sul gozzo, ci pensi?

Pronte ad essere tirate fuori dal computer dal primo sbirro abbastanza furbo da pensarci. O da ricorrere a qualche cervellone. Beh nessun problema, ora che lo sapevo. Leggere quella roba e cominciare a darmi da fare per sistemare la questione è stato tutt'uno. I vecchi messaggi e le altre istruzioni di Luthor, meglio cancellarli per davvero con uno di quei programmi. I file di swap, ripuliti. I messaggi successivi li avrei fatti con quel come-cazzo-si-chiama, con quel parametro del PGP, sì, -w. Nessuna traccia. Ma avevo un problema. C'era della roba che non potevo cancellare e basta. Certe liste per esempio. La mappa delle fogne con le indicazioni per ritrovare la Bomba. Allora, gagliarda l'idea di crittare una partizione del disco e ficcarci dentro tutto quello che scottava, eh? E pensa, che se uno non sa già che c'hai quella roba, non è che la vede ma non può leggerla e basta. Il bello è che non può neanche accorgersi che c'è. Quando avevo finito, il mio computer sembrava quello delle Dame di San Vincenzo alla tombola della parrocchia. Innocente come un neonato. Ed è stato un bene. Perché come avevo finito con quella faticaccia, ero stronco e non ce la facevo più a stare chiuso dentro quel cazzo di cottage. Uno lo capisce quand'è il momento di un po' di svago per sé e per i ragazzi, amico. È questa la differenza tra il vecchio Luthor e me. Un po' d'aria avrebbe fatto bene a tutti. Anche se Lex avrebbe dato fuori di matto a saperlo, chi poteva andarglielo a raccontare? E poi, lontani come eravamo da casa, chi ci poteva riconoscere? Anzi, tre omaccioni tutti soli asserragliati in un cottage senza mai uscire potevano dare nel naso a qualcuno più che se ogni tanto uscivano a svagarsi un po', mi spiego? Ragion per cui dico ai ragazzi di mettersi presentabili, lavarsi il collo eccetera che usciamo per vivere un po', ce lo siamo guadagnato, ecchecazzo. Com'è, come non è, quando torniamo al cottage, da lontano mi sembra di scorgere un chiarore dietro una finestra. Dura solo un attimo, ma mi basta. Dico ai ragazzi di non fare rumore. Ci avviciniamo di soppiatto alla porta ed entriamo con tutta l'artiglieria fuori e pronta al fuoco. E chi troviamo davanti al mio computer acceso? Ma Lois Lane e Jimmy Olsen, nientemeno! Come cazzo avevano fatto a sapere

che eravamo lì, dici? Bah, penso che la brava Lois avesse messo all'opera un po' del vecchio odor di figa con qualcuno dell'Organizzazione di Lex. Se non fossi di fretta, mi occuperei io di sapere con chi, credimi. Poi erano arrivati proprio quando ce ne stavamo uscendo e avevano preso l'occasione al volo. Naturalmente avevano solo perso tempo, perché dopo il lavoretto che aveva fatto il qui presente non c'era verso di trovare nulla nel mio computer. E non avevano ancora avvertito nessuno di essere lì, nemmeno Superman. La vecchia Lois è una che per tenersi uno scoop tutto per sé, venderebbe sua madre a un salsicciaio senza pensarci su due volte. E Jimmy non conta, fa sempre quello che dice lei. Naturalmente l'orologio di Jimmy, quello per chiamare Superman, è finito subito nel cesso. Ma che dovevo farne di quei due? Jimmy Olsen non parlava e cercava di darsi un'aria da duro. Però, con quell'aria da pretino irlandese uscito fresco fresco dal seminario, non gli riusciva mica troppo bene. Lois Lane invece strillava come un'aquila spennacchiata. Ha un bel caratterino quella. E la lingua lunga e affilata. Bah, poteva urlarmi che ero un nanerottolo orrido e un ratto di fogna puzzolente finché non si seccava la gola, per quanto mi riguardava. Quando le lingue lunghe cominciano a insultarti conviene lasciarle fare. Se sono lunghe abbastanza, finiscono sempre per lasciarsi scappare qualcosa. E infatti a un certo punto ha sbraitato che facevamo meglio a non torcerle nemmeno un capello, perché Superman lo sapeva che stavamo usando Internet, e avrebbe finito per beccarci lo stesso. Quando ha capito quello che si era lasciata scappare si è morsa le labbra e si è azzittita, ma era troppo tardi. Le ho spiegato paternamente: "Vedi, baby, forse tu non ce l'hai mica chiara la situazione. Al, lo vedi laggiù in quell'angolo, sono sei mesi che non tocca una donna. Prova a pensare all'effetto che gli fai in questo momento. Quanto a Louie, con lui quel tipo di rischio non lo corri mica. C'ha altri gusti, lui. Gli piace il rasoio, non so se mi spiego. Se il qui presente nanerottolo orrido gli dà il via, va a finire che spediamo a Superman le tue orecchie e il tuo naso in una busta. Magari anche capezzoli e clitoride, già che ci siamo. Così il tuo SuperCazzone avrà qualcosa per consolarsi pensando a te nelle lunghe notti d'inverno. Perché tanto non ti rivedrà mai più, mi spiego? Anche se poi ci becca, tu non sarai mica lì a goderti lo spettacolo. Se invece ora mi spieghi per benino quello che volevi dire, può anche darsi che riesci a riportare a casa intatta la tua bella pelle vellutata. Mi spiego?" Eccome se mi ero spiegato, amico. Per cui, mentre Jimmy Olsen le strillava di stare zitta, Lois ha vuotato il sacco tutto in una volta. Non che sapesse molto. Solo che Superman aveva messo su un fior d'apparecchiatura nella sua SuperCaverna, la Fortezza delle Seghe Solitarie, o come diavolo la chiama lui. E con quella stava cercando di rintracciarci sniffando le connessioni Internet. L'indirizzo di rete abituale di Luthor probabilmente lo conosceva già. Ma non gli serviva a molto, visto che a portare avanti tutta la storia era il qui presente. Ed ero io che avevo il pulsante della bomba, soprattutto. Ma gli sarebbe bastato monitorare tutti i messaggi ricevuti da Lex, salire a ritroso e scoprire i relativi mittenti, per ricavarne indicazioni utili su dove mi trovavo. Per uno sbirro normale sarebbe stato un bel casino. Corrompere decine di tecnici, ficcanasare tra montagne di byte... ma lo sai che Superman è bravino in questi giochetti. Lois ne sapeva anche meno di me ma aveva intuito che in quel modo, anche se non poteva capire quello che Lex e io ci raccontavamo, poteva sempre individua-

re da dove entravamo in rete. E c'era una cosa che non potevo nascondere, già. Il Pulsante della Bomba. Quello dovevo portarlo sempre addosso. Ero in un bel guaio amico. Dovevo avvertire subito Luthor, e quello che avevo da dirgli non gli sarebbe mica piaciuto tanto. E dovevo trovare anche il modo di farlo in modo meno diretto di come avevamo fatto finora, senza che Superman potesse risalire a me. Non mi restava altro che guardarmi un altro po' di quei fottuti capitoli, sperando di trovare qualcosa che mi servisse.

ANONYMOUS REMAILER

DI T.H.E. WALRUS

Per anonymous remailer si intende un particolare tipo di mail-server presente su Internet il cui compito è *ricevere e rinviare* un messaggio di posta elettronica oppure - con procedura leggermente diversa - un messaggio destinato a un newsgroup Usenet, in modo tale che dal messaggio, così come arriva al destinatario finale, sia impossibile risalire direttamente al mittente originario.

Basilarmente, l'anonymous remailer inoltra il messaggio a una qualunque destinazione successiva richiesta dal mittente, dopo aver effettuato una rimozione e sostituzione degli *header* (cioè dei vari campi "From:" "Return-Path:" "X-Sender" ecc.) che servono appunto per identificare chi è il mittente del messaggio. Inoltre - e questo è un aspetto particolarmente interessante dal nostro punto di vista - un corretto uso degli anonymous remailer, attraverso operazioni più complesse che verranno discusse in dettaglio, ci consente di intrattenere scambi di posta elettronica senza che terze parti in grado di "origliare" sulla rete possano controllarne il contenuto e\o tracciarne il percorso o anche solo venire a conoscenza della loro esistenza. Risulta inoltre possibile diffondere pubblicamente informazioni e notizie di ogni genere (ad esempio nei newsgroup o nelle mailing list) senza che eventuali avversari intenzionati a bloccarle possano rintracciare chi le diffonde.

L'impiego di queste tecniche di anonimato "forte" in Rete - a differenza di forme più "deboli", quali l'uso di alias, di account pubblici e via dicendo - garantisce un'effettiva irrintracciabilità e l'impossibilità di decifrazione della propria corrispondenza, anche da parte di organismi investigativi statali o sovranazionali dotati di risorse potenti. Ovviamente occorre farne uso con l'attenzione e la competenza del caso, peraltro non troppo difficile da raggiungere.

La ricerca di privacy e di anonimato può suscitare una certa perplessità e un certo sospetto, specie da parte di chi non conosce ancora bene la realtà della rete. Contro l'anonimato da tempo è in atto una massiccia campagna di opinione, supportata con forza da giornali e TV e sostenuta dai vari inevitabili esperti, magari appartenenti ad aree "democratiche e di sinistra". Assistiamo all'agitazione sempre più insistente e intimidatoria di rari (ma spesso gonfiati se non falsi) fatti di cronaca per gettare un'ombra su queste tecniche e su chi ne fa uso. Non si cerca di ammantare l'anonimato di semplice "illegalità" - che da un certo punto di vista è spesso difficile da provare e da un altro punto di vista sarebbe meno problematica per coloro a cui è destinato questo testo (almeno questa è la speranza) - ma si cerca di coprirlo di un'ombra di odiosità infamante.

Basta aprire un giornale, o guardare una trasmissione televisiva per accorgersene. La Rete pullula di mostri... Non c'è che l'imbarazzo della scelta: ci sono i mostri pedofili senza volto, in agguato negli angoli oscuri dei luoghi digitali, pronti a irretire, stuprare e uccidere le loro piccole innocenti vittime. E se non sono pedofili violenti, tenteranno almeno di corrompere irrimediabilmente i bambini mostrando loro pornografia e netsex...

Ci sono i mostri terroristi, che congiurano giorno e notte per massacrare a caso e insensatamente con le loro bombe i pacifici e inermi passanti. E l'attenta e paterna protezione assicurata ai bravi cittadini dagli eroici servitori dello Stato risulta spesso vana, impossibilitati come sono a sventare le oscure trame dei terroristi che sono possibili grazie all'anonimato dietro cui si celano...

Ci sono i mostri narcotrafficanti, che cinicamente usano anche la rete e l'anonimato che essa consente per aumentare l'osceno lucro del loro commercio di morte destinato ai giovani ingenui e sprovveduti...

Sono queste le figure che vengono martellate quotidianamente come spauracchi diabolici nell'immaginario collettivo da stampa, televisioni, convegni di esperti, conferenze di vescovi, discussioni nei bar e crocchi alle fermate dei tram, in un crescendo continuo di consenso automatico ed isterico. Perché questi sono i nuovi cavalieri dell'apocalisse, contro cui l'intera Civiltà Occidentale (sana, pacifica, giusta, religiosa e democratica per definizione) è chiamata a scendere in guerra e a difendersi. Ed è su questa santa guerra che deve forgiarsi il Nuovo Ordine Mondiale. Fu il Presidente Clinton in persona ad informarne l'umanità intera, rammentate?

Allora, via, per il sacrosanto obiettivo di smascherare questi demoni, cosa volete che sia una piccola rinuncia alla vostra privacy per il bene comune? ... Cosa? Non vi sta bene? Vi pare inaccettabile la rinuncia al vostro desiderio

di non rendere obbligatoriamente controllabile da chiunque ne abbia la debita Autorità quel che vorreste far sapere solo a pochi altri? Ma siamo tutti in guerra, lo sapete, e la crociata prosegue e si ingrossa... come mai non vi associate? Cos'è questa smania di voler restare anonimi? Ma cosa avete da nascondere?

La realtà, chiaramente, è un tantinello diversa. E capirlo e saperlo, anche per esperienza diretta non dovrebbe essere difficile, almeno per coloro a cui intende rivolgersi questo testo, al di là degli spauracchi terrorizzanti, degli stereotipi propagati dal linguaggio mistificante dello stato e del padrone, dei loro servitori e del gregge forcaiolo del consenso automatico ed entusiasta. Che il vero "terrorismo" sia spesso quello dello stato, che "pedofilia" (amore per bambini e adolescenti che non censura e rimuove la tensione erotica che essi possono vivere e suscitare) non significhi necessariamente stupro e violenza sull'infanzia - atti infami spesso invece operati nella famiglia e da coloro che se ne ergono a moralistici difensori - che il vero "narcotraffico" sia quello delle politiche proibizioniste e dagli enormi profitti "illegali" e "legali" che esse consentono, sono cose che in tanti abbiamo chiare. Anche se questi punti di vista vengono sempre censurati e rimossi dalla comunicazione sociale controllata o imbevuta di luoghi comuni.

E proprio per questo, siccome sono chiare per chi non voglia nascondersele, non avrebbe troppo senso discuterne qua ancora. Come non avrebbe troppo senso sottolineare che, una volta passata la risonanza mediatica di tante inchieste, una volta inflitta al capro espiatorio di turno un po' di galera "in attesa di giudizio" per fargli abbassare la cresta o il sequestro delle macchine per impedirgli di continuare la sua attività in rete, le successive assoluzioni o il cadere nel nulla delle accuse non vengono praticamente mai rese note...

Ma anche per chi queste cose le ha chiare, per chi conosce i meccanismi perversi dell'intreccio di connivenze e obiettivi che lega quasi in un unico sistema gli apparati mediatici, giudiziari e repressivi, resta spesso un residuo di dubbio, di perplessità... Perché usare queste tecniche? Perché voler essere anonimo, aumentando magari anche il rischio di essere criminalizzato, se di quel che sono e di quel che faccio non mi vergogno e anzi lo sbatto in faccia a chi vorrebbe farmene vergognare?

A questa domanda ci sarebbe una risposta semplice ed immediata: Perché - a parte la mia totale assenza di fiducia verso lo stato e le istituzioni - sono *io*, e solo *io*, che decido se e cosa far sapere di me ad altri. E che si tratti dei miei gusti in fatto di sesso o di preparazione di minestre, di politica o di marche di sigarette, di "droghe" o di barzellette, voglio poterne parlare a chi mi va e solo a lui. Senza che altri, siano essi il presidente Clinton, mia suocera

o qualche Supereroe possano metterci il naso se non voglio. E intendo fare tutto ciò che *so* per rendermelo possibile.

Dal mio punto di vista personale potrebbe bastare solo questa risposta. Ma c'è un'altra realtà più generale da tener presente. Non è facile rendersene immediatamente conto per chi non abbia familiarità con la Rete, ma *tutto* quel che viene immesso in essa è passibile di archiviazione, di conservazione indefinita e infine di *analisi*. Anche ciò che non sembrerebbe immediatamente "pubblico".

La minaccia è reale, e molto preoccupante. Sia chiaro: non è solo questa minaccia da tener presente, anche se chi si picca di antitecnologismo a tutti i costi nella Rete vedrà sempre e solo il Grande Fratello. Nella Rete è anche possibile appropriarsi, in tutti i sensi, di una grande ricchezza. Umana, e non solo di fredda informazione disponibile o di risorse di vario tipo da saccheggiare. Sono possibili infiniti incontri, suscettibili anche di sviluppi nella vita reale, al di là di ogni barriera, sia essa morale, di cultura dominante, di età, di sesso biologico, di limiti geografici, nazionalistici e burocratici. La rete è anche una realtà immensa di interazione capillare e continua in tempo reale, di cooperazione in atto. Su una scala planetaria mai raggiunta sinora e per ora *incontrollabile*. Un uso intelligente della rete consente a chi lo vuole di rendere note al mondo le proprie iniziative e opinioni, in barba ai divieti e alle censure che stati totalitari o democratici o gruppi di potere cercano di imporgli. E in Rete - mi va di dirlo qua - ho fatto alcuni degli incontri e delle esperienze più importanti e liberatorie della mia vita recente. Ma non mi nascondo che proprio per queste sue caratteristiche, la Rete può diventare anche uno degli strumenti di controllo e di dominio più potenti e distruttivi mai concepiti nella storia dell'umanità. Database di informazioni di ogni tipo, analisi globali sui gusti, le inclinazioni, le opinioni, gli acquisti di intere nazioni, di strati sociali o di singoli individui, disponibili per chi abbia il denaro e il potere per commissionarli. Identificazione del deviante. Controllo poliziesco capillare. Non è fantascienza. Già oggi è possibile e il tempo evolutivo della Rete è velocissimo. Andate a guardarvi la parte poliziesco-informatica degli accordi di Schengen o i programmi delle multinazionali per il controllo e la fidelizzazione del cliente o ancora le tecniche di spionaggio delle associazioni moralistiche, e capirete cosa intendo.

Capirete anche perché la battaglia, "tecnica" e "politica" insieme, per l'anonimato in rete è così importante. E capirete perché contro di esso, e contro le poche decine di server e le poche centinaia di uomini e donne che vi si impegnano (solitamente senza cavarne il minimo guadagno economico) elaborando software, diffondendo l'informazione, mettendo a disposizione siti e risorse per i servizi di anonymous remailing o di altro tipo, si accaniscono

in modo così apparentemente spropositato polizie, pennivendoli, mezzibusti televisivi, "esperti" e preti di tutti i colori e di tutto il mondo. E capirete perché siamo in tanti, in numero sempre crescente, a ritenere che sia necessario fare proprie queste tecniche, diffonderne le conoscenze di base e usarle anche se magari non abbiamo il famoso "nulla da nascondere".

Ma adesso basta. Per dirvi la mia su queste cose ce l'ho messa tutta. Se vi interessa, passiamo alla parte tecnica. Altrimenti, se non vi sentite a disagio a fare la parte del pesce rosso nella boccia di cristallo, probabilmente è inutile che io sprechi altro fiato.

Come trovare un anonymous remailer

Gli anonymous remailer di cui ci occuperemo in questa guida sono gratuiti e gestiti per lo più da appartenenti alla cosiddetta *comunità cypherpunk*, convinti assertori (per vari motivi) della necessità di assicurare questo servizio su Internet come aspetto fondamentale della difesa e tutela di anonimato e privacy in rete. Non deve quindi sorprendere che abbia luogo un ricambio abbastanza frequente: è tutt'altro che insolito che nel giro di pochi mesi o addirittura settimane alcuni anonymous remailer scompaiano e ne compaiano di nuovi. Sarebbe quindi abbastanza inutile fornire qui una lista di remailer, e anche se abbiamo avuto cura di verificare che tutti quelli citati nei nostri esempi fossero operativi al momento in cui scrivevamo, nulla assicura che lo saranno ancora al momento in cui questa guida verrà diffusa. Rimandiamo per questo alle varie liste che vengono frequentemente aggiornate su Internet, segnalando in particolare quella ormai canonica mantenuta da Raph Levien, la cui affidabilità è tale che alcuni tra i più diffusi programmi che automatizzano l'uso degli anonymous remailer si basano su di essa per aggiornare automaticamente il proprio elenco interno.

La pagina di Raph, oltre ad aggiornare automaticamente le liste dei vari tipi di remailer attivi su Internet e le loro chiavi pubbliche PGP, fornisce anche una serie di indicazioni preziose sulle loro opzioni di configurazione e particolarità di funzionamento - che come vedremo possono variare significativamente anche tra remailer dello stesso tipo - e sulla loro efficienza (tempo di risposta alla chiamata, eventuali periodi di down eccetera). In essa sono riportate anche varie notizie e link utili, che ne fanno un ottimo punto di partenza per l'esplorazione on-line del mondo degli anonymous remailer e dell'anonimato in rete.

Nei capitoli che seguono ci rifaremo alla ormai tradizionale classificazione "per tipo" degli anonymous remailer, anche se essa risulta ormai lievemente sorpassata (ad esempio i remailer di tipo *pseudoanonimo* sono ormai spariti dalla circolazione, mentre il tipo *mixmaster* accetta anche messaggi formattati nello stile cypherpunk). Tuttavia quasi tutta la documentazione disponibile su Internet si attiene a questa classificazione che consente anche una migliore chiarezza espositiva.

Anonymous remailer tipo 0 - detti anche *pseudoanonimi*, segnalati nella lista di Raph Levien come "penet". Riconoscono il campo "X-Anon-To:" come richiesta di remailing. Livello di sicurezza basso. Praticamente non più usati.

Anonymous remailer tipo 1 - detti anche *Cypherpunk*, segnalati nella lista di Raph Levien come "cpunk". Riconoscono il campo "Request-Remailing-To:" come richiesta di remailing. Livello di sicurezza elevato, se usati in catena e con crittazione PGP.

Anonymous remailer tipo 2 - detti anche *Mixmaster*, segnalati nella lista di Raph Levien come "mix". Accettano messaggi in formato proprietario elaborati con un apposito client. Il tipo di remailer più sicuro attualmente operativo su Internet.

Anonymous remailer pseudoanonimi (type 0)

Gli anonymous remailer di *tipo 0*, detti pseudoanonimi, hanno ormai un interesse più che altro storico. Questo tipo di remailer (chiamato anche "penet" dal più famoso di essi, anon.penet.fi , da tempo inattivo), forniva all'utente un account al quale veniva assegnato casualmente uno pseudonimo di identificazione. Il remailer manteneva un archivio segreto nel quale ogni pseudonimo era accoppiato al vero indirizzo di posta elettronica dell'utente. Nella posta in uscita gli *header* di identificazione del messaggio venivano sostituiti con lo pseudonimo. In questo modo era possibile per il destinatario finale del messaggio rispondere ad esso in modo facile e diretto utilizzando la funzione reply-to del proprio programma di posta elettronica; la posta in entrata destinata a uno pseudonimo veniva inoltrata al vero indirizzo e-mail corrispondente. A causa della facilità d'uso anon.penet.fi, mantenuto da Julf Helsingius, è stato per un lungo periodo il remailer più famoso e utilizzato di tutta Internet.

Purtroppo l'esistenza dell'archivio segreto di corrispondenze nomi-pseudonimi costituiva una notevole debolezza. L'operatore del sistema e i suoi

assistenti avevano accesso ai veri indirizzi degli utenti, la cui riservatezza dipendeva quindi dalla loro capacità e volontà di mantenerli segreti. A parte possibili indiscrezioni, la presenza del database rendeva anche il sistema particolarmente vulnerabile ad hackeraggi e intrusioni, teoricamente sempre possibili.

Questa debolezza intrinseca di anon.penet.fi divenne evidente in seguito a una denuncia per violazione di copyright presentata dalla Chiesa di Scientology. La nota setta californiana intendeva in realtà bloccare la diffusione di messaggi contenenti suoi testi interni, riservati ai propri adepti, ritenuti sacri e quindi non divulgabili. Helsingius, sentito in qualità di testimone, ricevette dalla polizia finlandese l'ordine di consegnare l'intero archivio degli utenti. In seguito al suo rifiuto nacque una causa legale rimasta celebre negli annali di Internet, al termine della quale egli riuscì a non consegnare l'intero archivio, ma fu comunque costretto a rivelare l'indirizzo e-mail dell'utente che aveva diffuso i messaggi. Anon.penet.fi non venne obbligato alla chiusura, e continuò a funzionare per un certo periodo dopo la vicenda, sino alla decisione personale di Helsingius di chiuderlo.

Anonymous remailer cypherpunk (Type I)

Questo tipo di anonymous remailer è molto più sicuro del precedente. Non esiste un archivio degli utenti e gli operatori dichiarano di evitare con la massima cura di tenere log (cioè registrazione dell'attività intercorsa sul server) che potrebbero permettere di identificare gli utenti. Inoltre un anonymous remailer cypherpunk dispone della propria chiave pubblica PGP ed è in grado di accettare posta crittata con essa, decrittarla e rinviarla all'indirizzo richiesto (alcuni remailer accettano solo posta crittata). Offre inoltre la possibilità di concatenare (*chaining*) il remailing attraverso diversi remailer in successione. Tipicamente, come vedremo in dettaglio, il concatenamento viene combinato con la crittazione PGP in modo da consentire una notevole sicurezza dell'anonimato. Gli anonymous remailer cypherpunk implementano inoltre funzioni quali il riordino casuale dei messaggi (*reordering*) e comandi per richiedere un ritardo nella trasmissione degli stessi (*latent time*) su cui ci si soffermerà un bel po' nei prossimi paragrafi a causa della loro importanza.

Sebbene i più recenti anonymous remailer di tipo mixmaster offrano garanzie ancora maggiori in fatto di sicurezza, il tipo cypherpunk è tuttora largamente diffuso su Internet, in quanto abbina ad un buon livello di sicurezza (se

usato in catena associata a crittazione) notevoli vantaggi dal punto di vista della gestione e dell'impiego. Ad esempio non è richiesto che l'operatore del remailer abbia un accesso di tipo "root" (cioè illimitato) sulla macchina dove gira il remailer stesso e soprattutto non è necessario alcun software particolare per la preparazione dei messaggi da inviare al remailer. Tutto quel che serve all'utente è la capacità di spedire un'e-mail, indipendentemente dal tipo di programma per la posta utilizzato o dal sistema operativo o ancora dalla macchina che ha a disposizione. Per questo motivo discuteremo come usare gli anonymous remailer cypherpunk in modo abbastanza particolareggiato.

A questo proposito avvertiamo però che la trattazione sarà limitata ai comandi base implementati da tutti i remailer di questo tipo. Ognuno di essi può disporre poi di funzioni particolari e comandi specifici, la cui conoscenza può risultare molto utile. Una prima informazione in questo senso è fornita dalla citata pagina Web di Raph Levien, ma per un uso estensivo di tutte le funzioni offerte da un particolare anonymous remailer è preferibile richiedere il suo file di help, cosa solitamente possibile indirizzando ad esso un'e-mail che abbia per Subject: remailer-help.

È bene inoltre sottolineare da subito che anche se presenteremo inizialmente, per ragioni di chiarezza espositiva, anche esempi dell'uso di un solo anonymous remailer e di concatenamento non crittato, a giudizio di chi scrive per scopi che vadano al di là del semplice divertimento *si raggiungono livelli di sicurezza accettabili solo con l'uso di almeno tre remailer in catena associato a crittazione PGP*. Le procedure per crittare e concatenare "a mano" possono apparire inizialmente un tantino complicate e scoraggianti, ma in realtà risulta assai più macchinoso spiegarle che non compierle, una volta che ci si sia impadroniti del meccanismo. Esistono vari programmi che consentono di semplificare - e in certa misura automatizzare - queste procedure (e su di essi daremo alcune informazioni in un successivo paragrafo di questa guida) ma è fondamentale, per un corretto utilizzo, avere la comprensione globale di ciò che si sta facendo. Invitiamo perciò a familiarizzare con le procedure "a mano" prima di passare all'uso di questi programmi, magari rieseguendo gli esempi che proponiamo.

Inviare un'e-mail attraverso un anonymous remailer cypherpunk

Si indirizza l'e-mail al remailer scelto. Deve contenere nella *prima linea* del corpo del messaggio solo i caratteri <::> (doppi due punti) senza nessuno spazio a sinistra. Nella linea immediatamente successiva deve essere inserita la stringa <Request-Remailing-To:>, seguita dopo uno spazio dall'indirizzo

a cui il remailer deve rinviare il messaggio. La stringa non deve lasciare spazi a sinistra e deve essere in questa *esatta* forma, comprese maiuscole e due punti. Deve poi intercorrere una linea bianca tra la stringa e l'inizio del testo. Da notare che diversi remailer supportano anche un formato abbreviato della stringa, tipicamente <Anon-To:>, ma il formato <Request-Remailing-To:> è quello canonico riconosciuto da tutti gli anonymous remailer di tipo cypherpunk, per cui faremo riferimento ad esso. Se dunque Joe (che per l'occasione scrive dal nuovo e immacolato indirizzo joe@freemail.net) deve comunicare con Lex Luthor usando il remailer "remailer@replay.com", dovrà preparare un messaggio così composto:

```
=====================================================
To: remailer@replay.com
From: joe@freemail.net
Subject: test
-----------------------------------------------------
::
Request-Remailing-To: lex@luthorcorp.com

Pare che il SuperFesso ci stia dando la caccia anche su Internet.
Sta controllando la posta che ricevi, ma passando attraverso un remailer
non riuscirà a risalire fino al nuovo indirizzo che sto usando.
=====================================================
```

Ricevuta questa e-mail l'anonymous remailer sostituisce gli header con i suoi e rinvia il messaggio all'indirizzo richiesto. Il messaggio arriverà a Luthor in questa forma (notare gli header "Return-Path:" e "From:") Notare anche che questo remailer non ha *strippato* (cioè eliminato) il subject:

```
============================================================
Return-Path: nobody@replay.com
Date: Sun, 27 Jul 1999 15:11:54 +0200 (MET DST)
Subject: test
To: lex@luthorcorp.com
From: nobody@replay.com (Anonymous)

Pare che il SuperFesso ci stia dando la caccia anche su Internet.
Sta controllando la posta che ricevi, ma passando attraverso un remailer
non riuscira' a risalire fino al nuovo indirizzo che sto usando.
============================================================
```

Inviare un'e-mail crittata attraverso un anonymous remailer cypherpunk

Se la posta in entrata sul remailer (o in uscita da Joe Lametta) è controllata, sarà buona cosa usare il PGP per impedire a terzi incomodi di leggere. Il

messaggio va crittato con la chiave pubblica dell'anonymous remailer e talora è obbligatorio farlo perché alcuni remailer non accettano posta se non è crittata.

Premettiamo al testo del nostro messaggio i doppi due punti <::> e nella linea immediatamente successiva il <Request-Remailing-To:> seguito dall'indirizzo del destinatario, lasciando poi una linea bianca, come fatto in precedenza per il messaggio non crittato:

```
::
Request-Remailing-To: lex@luthorcorp.com

Pare che il SuperFesso ci stia dando la caccia anche su Internet.
Sta controllando la posta che ricevi, ma passando attraverso un remailer
non riuscira' a risalire fino al nuovo indirizzo che sto usando.
```

e passiamo il tutto al PGP con la chiave pubblica del remailer a cui invieremo il messaggio, che in questo esempio sarà nuovamente "remailer@replay.com". In genere si ottiene la chiave pubblica del remailer richiedendo il file di help, ma ricordiamo anche che sulla solita pagina di Raph Levien sono disponibili le chiavi di tutti i remailer della lista. Il messaggio così diventa:

```
-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
Comment: PGP 2.x compatible

hIwC8cAPVx+T6qkBA/43Z7ATcC37Ip/+BGGlVEuqIGZu97QvbYeBDwd40i7ctXqa
8x6nqBdaj9Qwzur5scPzo5o510FPdt4XDxut7nMOUq3HbDg1i+O2WfNjXXYdJKS7
fbnpkxU9zKH0Suh8E4/imuDy9F2+7A5lBnX19Fx+ho8FsJ20e6YklUlzuw1FIKYA
AABZOlLMKedrpUbDxAwCXvz27ZmF/w05PL1ObJL81RXJQMVq7xnQtyZB5k+Tzuhr
9QsDWo4W73N3LdTRF6CNA1C6+zGIRKbQoyVt1c0e1bsh1Sh0nZI65zILaMs=
=kG6y
-----END PGP MESSAGE-----
```

A questo punto è necessario premettere al messaggio PGP i doppi punti <::> seguiti stavolta nella linea immediatamente successiva dalla stringa <Encrypted: PGP> senza nessun'altra aggiunta. Questa stringa avvisa il remailer che il messaggio è crittato. Come sempre, la stringa deve avere la forma esattamente come è riportata e deve intercorrere una linea bianca tra essa e il testo crittato:

```
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
Comment: PGP 2.x compatible

hIwC8cAPVx+T6qkBA/43Z7ATcC37Ip/+BGGlVEuqIGZu97QvbYeBDwd40i7ctXqa
8x6nqBdaj9Qwzur5scPzo5o510FPdt4XDxut7nMOUq3HbDg1i+O2WfNjXXYdJKS7
```

```
fbnpkxU9zKHOSuh8E4/imuDy9F2+7A5lBnX19Fx+ho8FsJ2Oe6YklUlzuw1FIKYA
AABZ0lLMKedrpUbDxAwCXvz27ZmF/wO5PLl0bJL81RXJQMVq7xnQtyZB5k+Tzuhr
9QsDWo4W73N3LdTRF6CNA1C6+zGIRKbQoyVt1cOe1bsh1ShOnZI65zILaMs=
=kG6y
-----END PGP MESSAGE-----
```

ora possiamo finalmente inviare il messaggio al remailer

```
=================================================================
To: remailer@replay.com
From: joe@freemail.net
Subject: test
-----------------------------------------------------------------
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i

hIwC8cAPVx+T6qkBA/43Z7ATcC37Ip/+BGGlVEuqIGZu97QvbYeBDwd4Oi7ctXqa
8x6nqBdaj9Qwzur5scPzo5o51OFPdt4XDxut7nM0Uq3HbDg1i+02WfNjXXYdJKS7
fbnpkxU9zKHOSuh8E4/imuDy9F2+7A5lBnX19Fx+ho8FsJ2Oe6YklUlzuw1FIKYA
AABZ0lLMKedrpUbDxAwCXvz27ZmF/wO5PLl0bJL81RXJQMVq7xnQtyZB5k+Tzuhr
9QsDWo4W73N3LdTRF6CNA1C6+zGIRKbQoyVt1cOe1bsh1ShOnZI65zILaMs=
=kG6y
-----END PGP MESSAGE-----
=================================================================
```

Il remailer ricevendo il messaggio leggerà la stringa <Encrypted: PGP> e decritterà il messaggio. A quel punto incontrerà la stringa <Request-Remailing-To:> messa in chiaro e provvederà al remailing all'indirizzo da essa indicato.

Inviare un'e-mail attraverso una catena di anonymous remailer cypherpunk (eventualmente crittata)

Finora abbiamo visto come rendersi anonimi utilizzando un singolo remailer. Per garantirci la sicurezza non solo che nessuno legga la nostra posta, ma anche che le nostre tracce si perdano tra remailer diversi, occorre utilizzare la crittazione in forma concatenata. Il concetto può sembrare a prima vista un po' ostico, ma qualche esempio semplificherà la comprensione.

Cominciamo considerando la *concatenazione* di più remailer. Si scelgono innanzitutto almeno tre anonymous remailer funzionanti del tipo Cypherpunk. Si invia l'e-mail al *primo* remailer della catena. Si inseriscono con le consuete modalità nella prima linea nel corpo del messaggio i doppi due punti <::> e

nella linea immediatamente successiva un <Request-Remailing-To:> seguito dall'indirizzo del *secondo* remailer. Si lascia una linea bianca. Si inseriscono nuovamente i doppi due punti e un <Request-Remailing-To:> seguito dall'indirizzo del *terzo* remailer e così via. L'*ultimo* <Request-Remailing-To:> deve essere seguito dall'indirizzo del destinatario finale (cioè di colui/colei che desidero riceva il messaggio):

```
=========================================================
To: remailer@replay.com
From: joe@freemail.net
Subject: test
---------------------------------------------------------
::
Request-Remailing-To: h_tuttle@rigel.cyberpass.net

::
Request-Remailing-To: remailer@anon.efga.org

::
Request-Remailing-To: lex@luthorcorp.com

Pare che il SuperFesso ci stia dando la caccia anche su Internet.
Sta controllando la posta che ricevi, ma passando attraverso un remailer
non riuscira' a risalire fino al nuovo indirizzo che sto usando.
=======================================================
```

Lex Luthor riceverà un messaggio di questo genere:

```
=================================================================
Return-Path: anon@server1.efga.org
Date: Sun, 27 Jul 1997 10:13:05 -0400
To: lex@luthorcorp.com
Subject: None
From: Anonymous <anon@anon.efga.org>

Pare che il SuperFesso ci stia dando la caccia anche su Internet.
Sta controllando la posta che ricevi, ma passando attraverso un remailer
non riuscira' a risalire fino al nuovo indirizzo che sto usando.
=================================================================
```

remailer@replay.com ha sostituito gli header e rinviato l'e-mail a h_tuttle@rigel.cyberpass.net che a sua volta ha sostituito e rinviato a remailer@anon.efga.org che da ultimo ha rinviato al destinatario finale. Lex Luthor può vedere solo la provenienza dall'ultimo remailer (notare che in questo caso uno dei remailer lungo il percorso ha strippato automaticamente il subject)

Una volta "disegnata la catena", cioè scelti i remailer a cui inviare il messaggio e la loro sequenza di rinvio, ci si preoccupa di crittare per ciascun

remailer il messaggio. Utilizziamo la catena dell'esempio precedente e procediamo passo per passo:

da joe@freemail.net a remailer@replay.com

joe@freemail.net invia a remailer@replay.com il messaggio, che è stato crittato consecutivamente con le chiavi PGP di tutti gli anonymous remailer della catena.

da remailer@replay.com a h_tuttle@rigel.cyberpass.net

remailer@replay.com deve ricevere il messaggio crittato con la sua chiave per poterlo decrittare - mettendo allo scoperto il messaggio crittato con le chiavi dei due remailer successivi e la richiesta di rinvio per h_tuttle@rigel.cyberpass.net

da h_tuttle@rigel.cyberpass.net a remailer@anon.efga.org

anche h_tuttle@rigel.cyberpass.net deve ricevere a sua volta il messaggio crittato con la sua chiave per poterlo decrittare e mettere allo scoperto il messaggio crittato con la chiave di remailer@anon.efga.org e la richiesta di rinvio ad esso

da remailer@anon.efga.org a lex@luthorcorp.com

remailer@anon.efga.org riceve il messaggio crittato con la sua chiave, lo decritta e infine rinvia a lex@luthorcorp.com

In altri termini il messaggio viene via "sbucciato" dalle crittazioni, sino al nocciolo finale costituito dal testo e dalla richiesta per l'ultimo remailing della catena di rinvio al destinatario finale. Dovremo quindi iniziare la crittazione dall'ultimo remailer, per poi procedere a ritroso crittando con le chiavi dei remailer che lo precederanno. In pratica cominciamo premettendo al testo la richiesta di remailing per lex@luthorcorp.com:

```
::
Request-Remailing-To: lex@luthorcorp.com

Pare che il SuperFesso ci stia dando la caccia anche su Internet.
Sta controllando la posta che ricevi, ma passando attraverso un remailer
non riuscira' a risalire fino al nuovo indirizzo che sto usando.
```

Passiamo quindi *tutto* (testo e richiesta di remailing) al PGP con la chiave dell'*ultimo* remailer della catena (remailer@anon.efga.org):

```
-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
```

```
hIwC8cAPVx+T6qkBA/43Z7ATcC37Ip/+BGGlVEuqIGZu97QvbYeBDwd40i7ctXqa
8x6nqBdaj9Qwzur5scPzo5o510FPdt4XDxut7nMOUq3HbDg1i+O2WfNjXXYdJKS7
fbnpkxU9zKH0Suh8E4/imuDy9F2+7A5lBnX19Fx+ho8FsJ20e6YklUlzuw1FIKYA
AABZOlLMKedrpUbDxAwCXvz27ZmF/w05PLlObJL81RXJQMVq7xnQtyZB5k+Tzuhr
9QsDWo4W73N3LdTRF6CNA1C6+zGIRKbQoyVt1c0e1bsh1Sh0nZI65zILaMs=
=kG6y
-----END PGP MESSAGE-----
```

Infine premettiamo i doppi due punti <::> e la stringa <Encrypted: PGP> che avviseranno remailer@anon.efga.org di decrittare il messaggio. A questo punto è pronto il "nocciolo interno" del messaggio, quello che sarà ricevuto da remailer@anon.efga.org, decrittato e trasmesso a lex@luthorcorp.com - *passaggio 4 della catena*:

```
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i

hIwC8cAPVx+T6qkBA/43Z7ATcC37Ip/+BGGlVEuqIGZu97QvbYeBDwd40i7ctXqa
8x6nqBdaj9Qwzur5scPzo5o510FPdt4XDxut7nMOUq3HbDg1i+O2WfNjXXYdJKS7
fbnpkxU9zKH0Suh8E4/imuDy9F2+7A5lBnX19Fx+ho8FsJ20e6YklUlzuw1FIKYA
AABZOlLMKedrpUbDxAwCXvz27ZmF/w05PLlObJL81RXJQMVq7xnQtyZB5k+Tzuhr
9QsDWo4W73N3LdTRF6CNA1C6+zGIRKbQoyVt1c0e1bsh1Sh0nZI65zILaMs=
=kG6y
-----END PGP MESSAGE-----
```

Ma come abbiamo visto questo messaggio sarà prima ricevuto da h_tuttle@rigel.cyberpass.net, che dovrà rinviarlo a remailer@anon.efga.org. Quindi dobbiamo nuovamente procedere a inserire una richiesta di remailing, stavolta a remailer@anon.efga.org:

```
::
Request-Remailing-To: remailer@anon.efga.org

::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
Comment: PGP 2.x compatible

hIwC8cAPVx+T6qkBA/43Z7ATcC37Ip/+BGGlVEuqIGZu97QvbYeBDwd40i7ctXqa
8x6nqBdaj9Qwzur5scPzo5o510FPdt4XDxut7nMOUq3HbDg1i+O2WfNjXXYdJKS7
fbnpkxU9zKH0Suh8E4/imuDy9F2+7A5lBnX19Fx+ho8FsJ20e6YklUlzuw1FIKYA
AABZOlLMKedrpUbDxAwCXvz27ZmF/w05PLlObJL81RXJQMVq7xnQtyZB5k+Tzuhr
9QsDWo4W73N3LdTRF6CNA1C6+zGIRKbQoyVt1c0e1bsh1Sh0nZI65zILaMs=
=kG6y
-----END PGP MESSAGE-----
```

Passeremo il tutto nuovamente al PGP, stavolta con la chiave di h_tuttle@rigel.cyberpass.net e infine premetteremo il solito <Encrypted: PGP>, ottenendo così il messaggio che arriverà a h_tuttle@rigel.cyberpass.net, verrà decrittato e sarà rinviato a remailer@anon.efga.org - *passaggio 3 della catena*

```
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
Comment: PGP 2.x compatible

hGwCXixrBPi1j4OBAwCscWd7X+GVfK4SJMGJxLv6/zuF9E6RYvsrE3qjcT5Vx1bX
Ibf5/kLlEnG5TQgg5CwI3fTShHUY+2ukocXn+C/U8P0LDOV56Pp0zStqsqs/8yVz
7m5e5W6ykWWVNyA5R3qmAAABuTxYZCLIDTeYvNhJNGN6LMS9XtZs1cFdG7TnevJS
b1iTgNiMnyRvkz2pi6NLMilWsNOVjRerDMAq984sN8pNMor9yI78oDu/FTcp4FYT
6B6n8HmYjTkKra7o5Sf00p6qW4I35bH8jguJrJZT6DBbmh3+YStq/b31NAuH+l21
agoVFX1fHbJdg5gMwxIb+dfeuE11sWKbPQpghGyxMK2pagar3+zq1W/Fgg6/YF6X
B0Mwy2g7Hhr92Mqrl9kgQGn6u3XWX51pIdJDj4vHJBpytH0oBqUHlW5ru32FHxEi
zkUDMmgq3O9tfBKka3zZewqFzgZb1d7isgXqBtDnqjq3ArDqIFT8XWHoPTsNGcuC
J72iIiZLqBce7zlpnaxmg1n0r+azaiXKNiZOhJBmXkgl38v8GEVCtKApjMgj9I7K
CDFPkh8CFPFcrEmdbqqASK3Wr/1tcvmGyGZ+yWkrXsB1gYvQh+mmJFCuRq8rUeJO
yJkB1TCAnVPQGLnvqMsKjJCgofToXbBVyNgbIWKPTdH2SZ16SP10aD4Vzg29vXcK
42PtNpBxtwB80ab9JmJDoLPd0e8LGyqrnL7irg==
=eD93
-----END PGP MESSAGE-----
```

Ancora una volta, dato che questo messaggio deve essere ricevuto e decrittato da remailer@replay.com per essere ritrasmesso a h_tuttle@rigel.cyberpass.net, inseriremo la richiesta di remailing a quest'ultimo:

```
::
Request-Remailing-To: h_tuttle@rigel.cyberpass.net

::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
Comment: PGP 2.x compatible

hGwCXixrBPi1j4OBAwCscWd7X+GVfK4SJMGJxLv6/zuF9E6RYvsrE3qjcT5Vx1bX
Ibf5/kLlEnG5TQgg5CwI3fTShHUY+2ukocXn+C/U8P0LDOV56Pp0zStqsqs/8yVz
7m5e5W6ykWWVNyA5R3qmAAABuTxYZCLIDTeYvNhJNGN6LMS9XtZs1cFdG7TnevJS
b1iTgNiMnyRvkz2pi6NLMilWsNOVjRerDMAq984sN8pNMor9yI78oDu/FTcp4FYT
6B6n8HmYjTkKra7o5Sf00p6qW4I35bH8jguJrJZT6DBbmh3+YStq/b31NAuH+l21
agoVFX1fHbJdg5gMwxIb+dfeuE11sWKbPQpghGyxMK2pagar3+zq1W/Fgg6/YF6X
B0Mwy2g7Hhr92Mqrl9kgQGn6u3XWX51pIdJDj4vHJBpytH0oBqUHlW5ru32FHxEi
zkUDMmgq3O9tfBKka3zZewqFzgZb1d7isgXqBtDnqjq3ArDqIFT8XWHoPTsNGcuC
```

CDFPkh8CFPFcrEmdbqqASK3Wr/1tcvmGyGZ+yWkrXsB1gYvQh+mmJFCuRq8rUeJO
yJkB1TCAnVPQGLnvqMsKjJCgofToXbBVyNgbIWKPTdH2SZ16SP10aD4Vzg29vXcK
42PtNpBxtwB80ab9JmJDoLPd0e8LGyqrnL7irg==
=eD93
-----END PGP MESSAGE-----

Passeremo il tutto al PGP con la chiave di remailer@replay.com e premetteremo l'avviso per la decrittazione, ottenendo così il messaggio che remailer@replay.com riceverà, decritterà e rinvierà a h_tuttle@rigel.cyberpass.net - *passaggio 2 della catena*:

```
::
Encrypted: PGP
```

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
Comment: PGP 2.x compatible

hIkCPRWysueuweUBA+dxD6rsXHFe1IJ1gGuVjGWPPDzE3DK3eE1yqgz6qDLQTblS
3P0tS1m0Hn21gBGPC22YucTYdPA1R5EDSbwX1bxq/yW+e20v/RdR8nlyGLUZrviZ
UAX5c2HXtfEImi82QleIsIReBLLtuNYHrnB07V0GKBKFoSElcpp38tbCtaYAAAMY
xpVeSNj5dRXudoYhUYdvOrZNjq/hqTkBLJcC1CEXizjlXofdW2LOntE2p+akkGtO
Uk8FNNsUBpipht2u3c/zya7zQbl6bEVm9dZkakEYhSkPw9BgDbAmjwamJv0toLC2
++Hiiv+2hPN1DNI73VeiaCyvN95Q9WkqFUro9DYghMGVCpdmlQW6D/BteegFRnEZ
/ihyJTocOmDP7EKmEoEditmEk8MedwYPE1xvNsFw1mdY9JmPRW61YW/MNfl8Izpo
D/xsjzchzxcMBqNiNq4D7V7s38VBdpjN0HoeMjtmRK0vVD7ROlaRtHspz3jnPdvj
q1zGZCLZGjell3QxNMGWcVhvDwcXOJgGn+Hx9l0zi6VyIwnMLJJrHIOxQ0wDld56
6KHdehpsHTjm4CH33Ia0jfCDHYHNY9Zoo/V4Q9p0vlSdiw05AUzWF3k7JS7i0ehJ
YlIwHI0xnxrjGdN7o0wt5HSAstL4aKTAIIKKJhxBwimM6yGdvnbmm8AVFZwnEiB6
a6MXrhHK5XWETTR3q/pVOgKfbvXjh4SpsdOLiezpD6Cr8joTrCphDCS3B+zlGTs3
xvs22yZXVK98EtgD7pnI1CRmsUtxMHjY7g8NpJwR6+WI/Y5JtvhLNJ2LcKDuTdFu
yZKyRTPj9Xl/A3QwD6uaW6kCjt5S7dtjiIXMCLlBI/AV9qUGUMk+Gm3TZI2d2/Pr
mJGs9em7pVktoBcmOzNdVE186+4tt/TQ187ic6UwbGZP0jrIhu2VrAWmLCI+vffS
iKFV8j19r96Y1V1V+i3fNjMfZa32/cka2KTW+efJdBxjtmJzqekDAxGJrQ64YKkV
jo6uAzvh0zXW0KobcrlhBgMFCIrrFZ+C/k5EX/Z9gBDxaALur7cu2ThsbYtCJXcT
64d+7Co3jP1WdiZyytiHfO4rSs0Ranm8NFMgjIuwILa2XnvL4I8JaxF1pqiBogHb
3b2+9SJVWVA7VpOH5ybmse+FWSzCRF021s8i1Y7H+4shyMZLwOImUjLK8haKMxvu
+/6v/sDlQKSHFf9GIzF9qaK/0JqbwIis
=YYCM
-----END PGP MESSAGE-----

E finalmente potremo spedire tutto quanto a remailer@replay.com - *passaggio 1 della catena*:

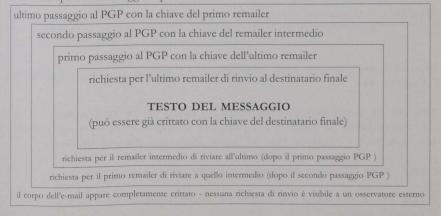
```
===========================================================
To: remailer@replay.com
From: joe@freemail.net
Subject: test
-----------------------------------------------------------
```

```
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
Comment: PGP 2.x compatible
```


```
hIkCPRWysueuweUBA+dxD6rsXHFe1IJ1gGuVjGWPPDzE3DK3eE1yqgz6qDLQTblS
3P0tS1m0Hn21gBGPC22YucTYdPA1R5EDSbwX1bxq/yW+e20v/RdR8nlyGLUZrviZ
UAX5c2HXtfEImi82QleIsIReBLLtuNYHrnB07V0GKBKFoSElcpp38tbCtaYAAAMY
xpVeSNj5dRXudoYhUYdvOrZNjq/hqTkBLJcC1CEXizjlXofdW2LOntE2p+akkGtO
Uk8FNNsUBpipht2u3c/zya7zQbl6bEVm9dZkakEYhSkPw9BgDbAmjwamJv0toLC2
++Hiiv+2hPN1DNI73VeiaCyvN95Q9WkqFUro9DYghMGVCpdmlQW6D/BteegFRnEZ
/ihyJTocOmDP7EKmEoEditmEk8MedwYPE1xvNsFw1mdY9JmPRW61YW/MNfl8Izpo
D/xsjzchzxcMBqNiNq4D7V7s38VBdpjN0HoeMjtmRK0vVD7ROlaRtHspz3jnPdvj
q1zGZCLZGjell3QxNMGWcVhvDwcXOJgGn+Hx9l0zi6VyIwnMLJJrHIOxQ0wDld56
yZKyRTPj9Xl/A3QwD6uaW6kCjt5S7dtjiIXMCLlBI/AV9qUGUMk+Gm3TZI2d2/Pr
mJGs9em7pVktoBcmOzNdVE186+4tt/TQ187ic6UwbGZP0jrIhu2VrAWmLCI+vffS
iKFV8j19r96Y1V1V+i3fNjMfZa32/cka2KTW+efJdBxjtmJzqekDAxGJrQ64YKkV
jo6uAzvh0zXW0KobcrlhBgMFCIrrFZ+C/k5EX/Z9gBDxaALur7cu2ThsbYtCJXcT
64d+7Co3jP1WdiZyytiHfO4rSs0Ranm8NFMgjIuwILa2XnvL4I8JaxF1pqiBogHb
3b2+9SJVWVA7VpOH5ybmse+FWSzCRF021s8i1Y7H+4shyMZLwOImUjLK8haKMxvu
+/6v/sDlQKSHFf9GIzF9qaK/0JqbwIis
=YYCM
```


```
-----END PGP MESSAGE-----
================================================================
```

In questo modo è possibile concatenare quanti remailer si vuole, sempre iniziando a crittare dall'ultimo anonymous remailer della catena e procedendo a ritroso. Nello schema grafico che segue dovrebbe a questo punto risultare chiaro come la sequenza dei passaggi al PGP risulti inversa rispetto alla sequenza di trasmissione mittente - remailer - destinatario finale. Le linee

Schema della concatenazione di tre anonymous remailer

<::> e <Encrypted: PGP> non sono state evidenziate nello schema in quanto hanno solo lo scopo di avvertire il remailer che il testo successivo ad esse è crittato.

Un particolare da ricordare: eventuale testo non crittato inserito dopo il testo PGP (ad esempio la firma che alcuni programmi per l'e-mail aggiungono automaticamente al momento della spedizione) sarà ritrasmesso e comparirà nel messaggio che arriva al destinatario finale.

Queste operazioni - piuttosto complesse da effettuare a mano - possono anche essere automatizzate con l'uso di specifici programmi, fra i quali ricordiamo Private Idaho per Windows (di cui parleremo più avanti), Premail per Linux e Chainmail per Macintosh.

Altri comandi per gli anonymous remailer cypherpunk

Cutmarks:

il comando <Cutmarks:> istruisce il remailer a rimuovere le linee di testo che cominciano con i caratteri specificati dal comando e quelle seguenti:

```
=====================================================
To: remailer@replay.com
From: joe@freemail.net
Subject:
-----------------------------------------------------

::
Request-Remailing-To: lex@luthorcorp.com
Cutmarks: **

Pare che il SuperFesso ci stia dando la caccia anche su Internet.
Sta controllando la posta che ricevi, ma passando attraverso un remailer
non riuscira' a risalire fino al nuovo indirizzo che sto usando.

Questa linea di testo comparirà nel messaggio
**
Questa linea di testo non comparirà nel messaggio
E nemmeno questa
=====================================================
```

<Cutmarks:> può essere usato in vari modi, ad esempio per non far comparire nel messaggio anonimo quelle linee che alcuni programmi di e-mail aggiungono automaticamente alla fine del messaggio (firma o altro) e che potrebbero portare all'identificazione del mittente. Notare l'assenza di linea bianca tra <Request-Remailing-To:> e <Cutmarks:>

Inserimento di header supplementari con il comando ##

Se inseriamo in una linea i caratteri <##>, nella linea di testo immediatamente successiva è possibile inserire degli header supplementari che compariranno nel messaggio anonimo. Come vedremo, in alcuni casi l'inserimento di un header particolare può essere necessario per postare un messaggio a un newsgroup Usenet. Questo comando può però esserci utile anche in altre occasioni, ad esempio se vogliamo assicurarci che un subject sia comunque visibile nel messaggio anonimo (ricordiamo che alcuni remailer strippano automaticamente il subject).

```
=======================================================
To: remailer@replay.com
From: joe@freemail.net
Subject: nessuno
-------------------------------------------------------

::
Request-Remailing-To: lex@luthorcorp.com

##
Subject: Superman vaffanculo

Pare che il SuperFesso ci stia dando la caccia anche su Internet.
Sta controllando la posta che ricevi, ma passando attraverso un remailer
non riuscira' a risalire fino al nuovo indirizzo che sto usando.
=======================================================
```

Notare la linea bianca tra <Request-Remailing-To:> e <##>.

Sicurezza contro modelli di minaccia avanzati

Gli anonymous remailer cypherpunk offrono garanzie molto migliori del vecchio tipo pseudoanonimo contro il ripetersi di episodi come quello di anon.penet.fi, specie se concatenati. Ma è evidente che la sicurezza offerta dall'invio di un'e-mail senza crittazione è comunque piuttosto bassa, specie se consideriamo che potrebbe esserci a nostra insaputa una registrazione di quanto noi trasmettiamo dalla nostra macchina al remailer. Anche l'impiego di un solo remailer associato a crittazione deve essere considerato a bassa sicurezza: il remailer conosce comunque sia il mittente che il destinatario e per quanto ne sappiamo potrebbe essere stato compromesso all'insaputa dell'operatore, oppure l'operatore stesso potrebbe aver configurato il remailer proprio allo scopo di bloccare e tracciare la posta altrui. L'ipotesi può parere un tantino paranoica, ma è proprio con un certo livello di paranoia razionale che occorre confrontarsi per valutare la sicurezza di queste tecniche.

La concatenazione degli anonymous remailer associata a crittazione offre una sicurezza molto più forte. Riconsideriamo il disegno della catena come nell'esempio precedente.

Joe - Remailer A. Joe trasmette ad A un messaggio crittato che solo A può decifrare. Un eventuale "origliatore" che controlli la trasmissione di posta da Joe ad A (o semplicemente tutta la posta in ingresso nel remailer A) potrà sapere solo che Joe ha spedito un'e-mail ad A. Non conosce né il contenuto di essa, né il destinatario finale, né i successivi remailer della catena.

Remailer A - Remailer B. A conosce Joe, ma una volta decifrato il messaggio sa solo che deve rinviarlo a B che a sua volta è l'unico che può decifrare il testo PGP dopo che esso è stato sottoposto alla prima decrittazione da parte di A.

Remailer B - Remailer C. B non sa più che Joe è il mittente originario, sa solo che il messaggio proviene da A e che, una volta decrittato, deve rinviarlo a C.

Remailer C - Lex. C sa solo che il messaggio gli arriva da B. Una volta decrittato il messaggio conosce il destinatario finale (Lex) e il contenuto del messaggio (ma solo se il messaggio stesso non è ulteriormente crittato con la chiave pubblica del destinatario); non conosce né il mittente né il percorso del remailing precedente a B.

Di conseguenza, per ricostruire il percorso mittente-destinatario bisognerebbe che tutti i remailer della catena fossero compromessi. È quindi

sufficiente che uno solo dei remailer sia sicuro per mantenere l'integrità del nostro anonimato. È inoltre evidente che tanto più lunga è la catena tanto più alta è la probabilità che almeno un remailer sia sicuro, anche se considerazioni pratiche sulla garanzia dell'arrivo a destinazione del messaggio limitano solitamente questa lunghezza a poco più di tre remailer.

Potrebbero esistere però forme di attacco alla sicurezza del nostro anonimato estremamente più sofisticate e potenti. Per dare un'idea di esse faremo riferimento al "modello di minaccia" proposto da Lance Cottrell (l'autore di Mixmaster) per valutare la sicurezza degli anonymous remailer. Secondo Cottrell chi cerca di intercettare i nostri messaggi (l'attaccante) potrebbe disporre di una serie di capacità da non sottovalutare. Ipotizziamo cinque situazioni:

1) l'attaccante ha compromesso alcuni remailer (ma non tutti) e conosce punto di partenza, destinazione e contenuto di tutti i messaggi che passano attraverso gli anonymous remailer compromessi;

2) l'attaccante può monitorare tutti i messaggi al momento in cui lasciano la macchina del mittente originario e al momento in cui arrivano alla macchina del destinatario finale;

3) l'attaccante è in grado di monitorare tutti i messaggi in entrata e in uscita da tutti gli anonymous remailer, nonché l'ora di arrivo e di partenza;

4) l'attaccante può impedire ai messaggi di giungere a destinazione (*denial of service attack*);

5) l'attaccante è in grado di inviare un numero illimitato di messaggi attraverso gli anonymous remailer (*spam attack*) inclusi quelli precedentemente registrati (*reply attack*).

Ad evitare paranoie irrazionali, è bene sottolineare che questo modello teorico di minaccia fa riferimento, anche a detta dello stesso Cottrell, al classico "peggior scenario possibile" e ipotizza un attaccante estremamente potente e determinato a un impiego di risorse tale da risultare abbastanza improbabile e antieconomico, specie rispetto alle situazioni per cui facciamo comunemente uso delle tecniche per l'anonimato in rete.

Ma è bene sottolineare anche che l'attuazione pratica di questo modello non è affatto impossibile, specie se pensiamo ad esempio alle risorse di cui potrebbero eventualmente disporre agenzie governative o sovranazionali fortemente motivate a monitorare lo scambio in rete tra alcuni soggetti, o magari a rintracciare chi stia diffondendo in rete notizie particolarmente critiche.... Comunque sia, se l'attaccante dispone solo delle risorse di cui ai punti uno e due, abbiamo già visto come la tecnica di concatenamento

insieme a quella di crittazione degli anonymous remailer cypherpunk sia già sufficiente a mantenere l'integrità del nostro anonimato a livelli più che accettabili. Ma se dispone anche della capacità di cui al punto tre (monitoraggio dei messaggi in entrata e in uscita da tutti i remailer e registrazione dell'ora) allora le cose potrebbero cambiare: se il nostro messaggio venisse rinviato in uscita dal remailer subito dopo l'arrivo del corrispondente messaggio in entrata, l'attaccante potrebbe correlare i due messaggi e procedere così al tracciamento lungo tutta la catena dei remailer. Per questo motivo ormai tutti i remailer di tipo cypherpunk avanzato implementano una particolare funzione denominata reordering (questa funzione è svolta automaticamente dal remailer e non richiede comandi da parte dell'utente). Il tipo più semplice di reordering viene effettuato conservando ogni messaggio sul remailer per un periodo di tempo casuale prima di procedere al rinvio, eliminando attraverso questo ritardo casuale (detto latency - da non confondere con il comando <Latent-Time:>) la possibilità di correlare i messaggi in uscita a quelli in entrata basandosi sull'ora d'arrivo e partenza.

Tuttavia, la protezione offerta da questo tipo di reordering dipende in gran parte dal volume di traffico che passa attraverso il remailer. Se per qualche motivo dopo l'arrivo del nostro messaggio non ne arrivassero altri per un periodo di tempo più lungo di quello massimo previsto per il ritardo casuale - come potrebbe accadere in seguito al blocco dei successivi messaggi in arrivo da parte dell'attaccante (*denial service attack*, punto quattro del modello di minaccia) o anche solo per casuali fluttuazioni di traffico o problemi sulla rete - allora il messaggio rinviato in uscita sarebbe facilmente correlabile all'ultimo messaggio in entrata. E per ragioni di praticità la durata massima dell'intervallo casuale non può essere prolungata più di tanto

Una tecnica di reordering più evoluta prevede invece che un determinato numero di messaggi, il cosiddetto "pool", venga sempre e comunque mantenuto in giacenza sul server. All'arrivo di un nuovo messaggio viene rinviato un messaggio in uscita scelto a caso tra tutti quelli giacenti nel pool, incluso quello appena arrivato. Questa tecnica - che può essere combinata a quella del ritardo casuale per una sicurezza ancora maggiore - è però anch'essa vulnerabile se l'attaccante possiede la capacità di inviare al remailer un gran numero di messaggi, superiore a quello dei messaggi che di default sono mantenuti nel pool (*spam attack*, punto cinque del modello di Cottrell). In seguito a questa massiccia ondata di arrivi, tutti i messaggi giacenti nel pool saranno rilasciati e sostituiti da quelli inviati dall'attaccante e a esso noti. All'arrivo del nostro messaggio l'attaccante farà seguire un ulteriore spam, provocando nuovamente il rinvio di tutti i messaggi e a questo punto gli saranno tutti noti tranne il nostro, che verrà così identificato e tracciato.

Un'ulteriore protezione contro questo tipo di attacchi ci è però offerta dal comando <Latent-Time:>. Con questo comando l'utente può richiedere al remailer che il suo messaggio venga in ogni caso trattenuto prima del rinvio, per un tempo specificato dall'utente stesso, indipendentemente dal reordering effettuato dal remailer stesso.

A causa della sua importanza daremo qui un esempio delle varie possibilità del suo impiego:

```
To: remailer@replay.com
From: joe@freemail.net
Subject: test
------------------------------------------------------

::
Request-Remailing-To: h_tuttle@rigel.cyberpass.net
Latent-Time: +6:00

::
Request-Remailing-To: remailer@anon.efga.org
Latent-time: +3:15r

::
Request-Remailing-To: lex@luthorcorp.com
Latent-Time: 17.30

Pare che il SuperFesso ci stia dando la caccia anche su Internet.
Sta controllando la posta che ricevi, ma passando attraverso un remailer
non riuscira' a risalire fino al nuovo indirizzo che sto usando.
```

In questo esempio di concatenamento non crittato il primo <Latent-Time:> chiede a remailer@replay.com - a cui è indirizzato il primo messaggio della catena - un ritardo di 6 ore prima di procedere al remailing a h_tuttle@rigel. cyberpass.net. Notare il "+", l'assenza di spazi tra il "+" e la durata del ritardo e il formato con cui si indica la durata del ritardo in <ore>:<minuti> (durata massima 24 ore). Notare inoltre l'assenza di linea bianca tra Request-Remailing-to: e Latent-Time:

Il secondo <Latent-Time:> chiede a h_tuttle@rigel.cyberpass.net di ritardare il rinvio a remailer@anon.efga.org per un intervallo di tempo casuale non superiore a 3 ore e 15 minuti, indicato dalla lettera "r". Infine il terzo Latent-Time: indica a remailer@anon.efga.org di effettuare il rinvio a lex@luthorcorp.com alle 5 e mezzo del pomeriggio (ora locale). In questo caso l'ora non è preceduta dal "+", e deve essere specificata nel formato delle 24 ore.

Nella pratica reale, i vari comandi <Latent-Time:> verranno specificati per ogni singolo remailer della catena via via che si procede nei vari passaggi della crittazione. Dovrebbe risultare chiaro da quanto visto sinora che il raf-

forzamento della sicurezza offerto da questo comando sarà tanto maggiore quanto più lungo sarà il ritardo richiesto.

In conclusione, la sicurezza offerta dagli anonymous remailer cypherpunk, grazie alla tecnica di concatenamento associato a crittazione PGP, irrobustita dalle funzioni di reordering dei remailer e dall'uso del comando <Latent-Time:> risulta molto forte. In particolare risulta in grado di resistere anche alla compromissione di quasi tutti gli anonymous remailer della catena, al monitoraggio diretto della macchina da cui partono i messaggi e ad attacchi sofisticati che comportano la disponibilità di ingenti risorse, quali quelli basati sulla registrazione dell'ora di arrivo e partenza di tutti i messaggi che transitano attraverso la rete dei remailer.

La tecnica finora proposta potrebbe risultare teoricamente vulnerabile di fronte a tipi di attacco ancora più potenti di quelli appena visti, quali quelli basati sull'*analisi della dimensione dei messaggi.* Anche se il nostro messaggio è correttamente concatenato, crittato, riordinato dagli anonymous remailer della catena e ritardato su ognuno di essi con il comando <Latent-Time:>, esso comunque alla partenza è costituito da un numero preciso di byte - difficilmente identico a quello di un altro - che diminuirà a ogni successivo passaggio della catena in modo prevedibile e conosciuto. Un attaccante che disponga delle risorse di cui al modello di Cottrell e di una notevole capacità di calcolo potrebbe tentare di identificarlo e tracciarlo basandosi proprio su questa caratteristica. Come contromisura sarebbe possibile pensare di inserire nel nostro messaggio delle "imbottiture", cioè del testo-spazzatura che i remailer potrebbero rimuovere ad ogni rinvio (cfr. comando <Cutmarks:>) rendendo così imprevedibile il rapporto tra la dimensione del messaggio in entrata con quella in uscita. Questa soluzione presenta però l'inconveniente di aumentare sensibilmente le dimensioni del messaggio, spesso sopra i limiti di accettazione degli anonymous remailer, in quanto, per essere veramente efficace, bisognerebbe che il messaggio cambiasse di dimensione per una grossa frazione ad ogni balzo.

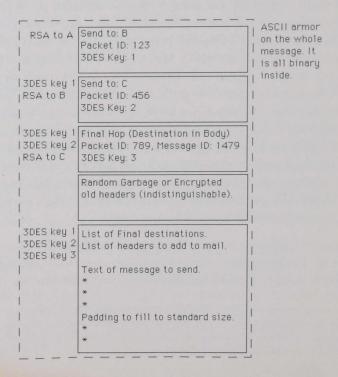
Inoltre, essa potrebbe offrire informazioni a un attaccante che abbia compromesso il remailer: dato che questa soluzione è poco usuale, il nostro messaggio potrebbe benissimo essere il solo che presenta una diminuzione del numero di byte diversa da quella standard. La migliore soluzione sarebbe quella che tutti i messaggi avessero la stessa dimensione.

Un'altra possibilità di attacco potrebbe basarsi sulla tecnica del cosiddetto *reply attack*: l'attaccante registra il nostro messaggio al momento in cui lascia la nostra macchina e ne invia un gran numero di copie al primo remailer della catena. Un gran numero di messaggi identici verranno così rinviati al remailer successivo, che verrà identificato attraverso questo improvviso ed

anomalo aumento del traffico tra i due remailer, ben visibile ad un'analisi basata sul volume dello stesso. In questo modo sarebbe possibile seguire il percorso del nostro messaggio sino al destinatario finale. Per questo motivo sarebbe opportuno che gli anonymous remailer non accettassero di rinviare più di una volta lo stesso messaggio. Come vedremo, i più recenti remailer del tipo *mixmaster* offrono le migliori garanzie contro questo tipo di attacchi.

ANONYMOUS REMAILER TIPO MIXMASTER (TYPE II)

Mixmaster è il tipo più recente e sicuro di remailer attualmente operativo su Internet e rappresenta "lo stato dell'arte" nel campo dell'anonymous remailing.

Concatenamento e crittazione sono predisposti in modo automatico al momento della preparazione del messaggio con l'apposito client. Mixmaster non usa la crittazione PGP ma il pacchetto RSAREF e alcuni formati proprietari in modo da implementare un processo di crittazione\decrittazione estremamente più complesso di quello "a cascata" degli anonymous remailer cypherpunk, come si può immediatamente intuire dal seguente schema grafico del pacchetto mixmaster, elaborato da Cottrell stesso.

In pratica mixmaster scompone i messaggi, li sottopone a crittazione multipla e li incapsula in uno o più pacchetti di dati di uguale dimensione, rendendo così impossibile un'analisi efficace basata su questa caratteristica. I pacchetti vengono inviati separatamente lungo la rete dei remailer e sottoposti a un sofisticato reordering. Ogni pacchetto è interamente crittato a ogni rinvio con una chiave 3DES, in modo che nessuna informazione sia visibile a un osservatore esterno. Come per gli anonymous remailer cypherpunk, ogni remailer della catena, anche se compromesso, può conoscere solo il punto da cui arriva un pacchetto e la destinazione di rinvio, ma oltre a questo, soltanto l'ultimo remailer della catena è in grado di vedere quali pacchetti compongono un singolo messaggio, mentre per tutti gli altri essi sono completamente indipendenti tra loro. A ogni pacchetto di dati in transito viene casualmente assegnato un numero di identificazione che il remailer conserva in memoria. Il sistema è così in grado di riconoscere un pacchetto già transitato e rifiutarne il rinvio, proteggendosi da un possibile "reply attack" basato sulla cattura e ritrasmissione dei pacchetti. Infine, come ulteriore protezione, l'architettura di un remailer mixmaster prevede un traffico costante "di copertura" generato casualmente, in modo da nascondere la trasmissione dei pacchetti reali entro un "rumore di fondo" variabile.

A giudizio di chi scrive, questi notevoli miglioramenti sono però in certa misura controbilanciati dal fatto che Mixmaster, a differenza degli anonymous remailer tipo Cypherpunk, necessita di un apposito client per la preparazione dei messaggi, non utilizzabile sotto ogni sistema operativo. Al momento in cui scriviamo il client è disponibile per UNIX, in un unico pacchetto di distribuzione assieme al server. Le release più recenti di questo pacchetto contengono anche i sorgenti e le istruzioni per la compilazione del client sotto DOS, operazione però non facilissima per l'utente medio. Esiste anche una vecchia release del client per DOS, che tuttavia non funziona come applicazione *stand-alone* (cioè funzionante da sola) ma è disegnata per essere installata come applicazione aggiuntiva per Private Idaho. Private Idaho è probabilmente il miglior programma per l'uso integrato di PGP, anonymous remailer cypherpunk o mixmaster e nym server. È disponibile gratuitamente per Windows 3.11, per Windows 95/98, per Windows NT e i sorgenti sono pubblicamente disponibili[1].

Questo svantaggio è in parte compensato dal fatto che gli anonymous remailer di tipo mixmaster possono accettare anche messaggi nel formato cypherpunk. Se nella catena il remailer successivo è anch'esso un mixmaster,

[1] Oltre a questa versione del client mixmaster (mixmaste.zip) è disponibile una nuova release (mix204b1.zip) utilizzabile invece sia sotto DOS sia sotto Private Idaho. In entrambi i casi occorre prelevare dalla rete il client. Non esistono attualmente versioni per altri sistemi operativi, anche se dovrebbe essere in preparazione una versione per Macintosh.

allora i messaggi gli verranno comunque rinviati in forma "incapsulata", anche se non sarà possibile per l'utente disporre delle piene funzionalità mixmaster.

Inviare un'e-mail attraverso gli anonymous remailer mixmaster

La prima cosa da tener presente è che è assolutamente necessario aggiornare ogni volta la lista degli anonymous remailer mixmaster, le loro chiavi proprietarie e - utilissimo, anche se non indispensabile - le statistiche sul loro stato di funzionalità. Questa procedura, come quella di invio dei messaggi, risulta assai più macchinosa e a rischio di errore con il client *stand-alone* per DOS che non nell'uso assieme a Private Idaho. È consigliabile pertanto per gli utenti DOS usare sempre Private Idaho per l'impiego del client mixmaster, a meno che non vi siano sostanziali e validi motivi per non usare Windows (ad esempio i non vedenti possono trovarsi in difficoltà con l'uso di interfacce grafiche, oppure si può disporre di macchine talmente obsolete da non consentire l'uso di Windows 3.11 o superiori).

Qui daremo solo le indicazioni strettamente necessarie per l'installazione e l'uso del client mixmaster mix204b1.zip con Private Idaho versione 2.8. Entrambi questi software sono disponibili all'URL http:// www.ecn.org/ crypto/. L'help completo di Private Idaho 2.8 è disponibile sul sito dell'autore, Joel McNamara. Esiste anche una versione italiana, redatta da Putro, anch'essa reperibile all'URL http:// www.ecn.org/crypto/.

Installazione

Decomprimiamo i file del pacchetto e muoviamoli nella directory contenente Private Idaho, solitamente chiamata c:\pidaho. I file dumpout.exe e htm2lis.exe non sono richiesti per l'uso con Private Idaho. Aggiungere al proprio autoexec.bat la linea "mixpath=c:\pidaho", specificando eventualmente il percorso completo di Private Idaho se è diverso. A questo punto si deve spegnere e riavviare il proprio computer per rendere operativa la variabile d'ambiente (cioè la modifica fatta all'autoexec.bat).

Aggiornamento delle liste

Questa procedura è indispensabile, in quanto il numero dei remailer Mixmaster attivi su Internet è soggetto a variazioni e le loro chiavi pubbliche vengono cambiate periodicamente. Ovviamente le liste fornite assieme a Private Idaho saranno ampiamente sorpassate al momento in cui installerete il client, quindi è consigliabile procedere all'aggiornamento delle liste ogni

qualvolta si desidera inviare un messaggio attraverso la rete dei remailer mixmaster. Bisogna innanzitutto connettersi in rete e lanciare Private Idaho. Selezionare "Mixmaster" dal menu "Remailer" e fare click su "Update remailer info" dallo stesso menu. Private Idaho si connetterà automaticamente alle pagine WWW che contengono i dati per l'aggiornamento e lo effettuerà senza bisogno di ulteriori interventi. Comparirà una finestrella che informa sull'andamento del processo sino alla conclusione.

Uso di Mixmaster con Private Idaho

Non differisce da quello degli anonymous remailer Cypherpunk con Private Idaho. Mentre si è off-line si scrive il messaggio nella finestra di testo e lo si indirizza al destinatario finale. Si seleziona "Mixmaster" dal menu "Remailer", comparirà nella finestra "remailer names" l'elenco dei remailer mixmaster con le statistiche di affidabilità e l'indicazione della data dell'ultimo aggiornamento. È possibile selezionare direttamente un solo remailer, ma è preferibile selezionare "chain" per concatenare. Facendo click con il mouse su "Append Info" dal menu "Message", compare la finestra "Remailer Chain". Selezionare i remailer desiderati. Private Idaho preparerà il messaggio per spedirlo all'anonymous remailer Mixmaster: al termine di queste operazioni ci si può connettere in rete e fare click sul tasto "send" per spedire il messaggio.

Gli utenti Linux, (e chi desiderasse usare il client Mixmaster come applicazione *stand-alone* in ambiente DOS) possono fare a meno di Private Idaho e usare direttamente *il client in modalità interattiva*. Per questa possibilità - come pure per quella di configurare varie funzionalità del client - è meglio leggere attentamente la documentazione acclusa alle relative distribuzioni.

Postare un messaggio in un Newsgroup Usenet in maniera anonima

In primo luogo è possibile postare in un Newsgroup direttamente attraverso un anonymous remailer, ma solo pochissimi remailer offrono accesso a questo servizio, che aumenta notevolmente volume di traffico e carico gestionale. Gli anonymous remailer tipo Cypherpunk che offrono questa possibilità direttamente sono indicati come "post" nella mai abbastanza lodata pagina di Raph Levien, ma è bene assicurarsene direttamente contattando il remailer e richiedendo l'help.

La procedura è simile a quella precedentemente descritta, cambia soltanto la stringa della richiesta di rinvio, che deve essere nel formato <Anon-

Post-To:> seguito dal nome del Newsgroup. Dato che per accedere a un Newsgroup è necessario avere un Subject (che abbiamo visto potrebbe essere cancellato lungo il percorso) ne inseriremo anche uno addizionale, ricorrendo al noto comando <##>:

```
=====================================================
To: remailer@replay.com
From: joe@freemail.net
Subject: none
-----------------------------------------------------

::
Anon-Post-To: alt.metropolis

##
Subject: annuncio alla cittadinanza

Cari concittadini: avete una bomba termonucleare sotto il culo. I
poveri Lois Lane e Jimmy Olsen ci sono legati vicino. Superman si
puo' fottere.
=====================================================
```

È naturalmente possibile fare di questo messaggio l'ultimo di una catena. In questo caso ovviamente, la richiesta di rinvio rivolta agli anonymous remailer precedenti l'ultimo dovrà essere nel consueto formato <Request-Remailing-To:> ed essi non dovranno necessariamente supportare la possibilità di inviare un messaggio su Usenet. Per un'analoga possibilità di invio diretto su Usenet con i remailer tipo Mixmaster, è utile consultare la documentazione inclusa nella distribuzione del client.

Un'alternativa molto più frequente è quella di utilizzare un *gateway usenet*, cioè un servizio che provvede a trasferire un messaggio di posta elettronica a un Newsgroup. Da notare che questi gateway *non sono un servizio anonimo* e quindi sarà necessario far passare la nostra e-mail attraverso un anonymous remailer o meglio ancora una catena che provvederà al rinvio al gateway stesso (che a sua volta invierà al newsgroup).

Esistono due tipi di gateway per Usenet, il tipo *e-mail gateway* che legge il newsgroup a cui inviare il messaggio direttamente dall'indirizzo inserito nel messaggio e il tipo *mail2news*, che legge un *header* addizionale da inserire nel messaggio stesso. Un *e-mail gateway* possiede un indirizzo del tipo "group.name@uni-stuttgart.de". Per postare è necessario spedire un e-mail a quell'indirizzo, sostituendo a "group.name" il nome del newsgroup e lasciando inalterato il resto. La richiesta di remailing per l'anonymous remailer avrà quindi la solita forma:

```
=========================================================
To: remailer@neva.org
From: joe@freemail.net
Subject: none
---------------------------------------------------------

::
Request-Remail-To: alt.metropolis

##
Subject: annuncio alla cittadinanza

Cari concittadini: avete una bomba termonucleare sotto il culo. I
poveri Lois Lane e Jimmy Olsen ci sono legati vicino. Superman si
puo' fottere.
=========================================================
```

I gateway stile *mail2news* hanno invece un indirizzo fisso, ad esempio "mail2news@myriad.alias.net". Per inviare un messaggio è necessario spedirlo a quell'indirizzo *senza variazioni*, aggiungendo invece l'*header supplementare* "Newsgroups:" seguito dal nome del Newsgroup (o di più newsgroup separati da una virgola senza lasciare spazi):

```
=========================================================
To: remailer@neva.org
From: joe@freemail.net
Subject: none
---------------------------------------------------------

::
Request-Remail-To: mail2news@myriad.alias.net

##
Subject: annuncio alla cittadinanza
Newsgroups: alt.metropolis,alt.superman.fans

Cari concittadini: avete una bomba termonucleare sotto il culo. I
poveri Lois Lane e Jimmy Olsen ci sono legati vicino. Superman si
puo' fottere.
=========================================================
```

Preparare un indirizzo anonimo con un reply-block

Intendiamo per indirizzo anonimo un indirizzo a cui possiamo ricevere posta senza che chi ce la spedisce conosca il nostro nome o il nostro vero indirizzo di posta elettronica. Anche se per questo tipo di servizio esistono dei tipi particolari di server, chiamati Nym Server, di cui si parlerà nel prossimo capitolo, è possibile utilizzare anche gli anonymous remailer Cypherpunk per preparare un cosiddetto *reply-block*, cioè un file crittato con la chiave pubblica di uno o più remailer contenente una richiesta di rinvio al nostro vero indirizzo di posta elettronica. Prepariamo un file di testo con la ormai nota richiesta di rinvio al nostro vero indirizzo:

```
::
Request-Remailing-To: joe@freemail.net
```

passiamo al PGP con la chiave pubblica di un remailer, ad esempio "remailer@anon.efga.org" e premettiamo i soliti doppi due punti seguiti da "Encrypted: PGP"

```
::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
Comment: PGP 2.x compatible

hIwC8cAPVx+T6qkBBACY8/z0p83CZJDeSrb9N1f0hAJLxgcwSjtX6f6Y7dFHUdp8
YpM1IOCSEcZBPwcW1CPMnv9/xkQFhYFLRV9s0ksiw411dPgtKg3YF8h4ZRwdfecp
VcFZD2qqN7UJa/cGp4nASXA/PQmBkgRJcvts3v8Lt4qCaxTFVg0EY4WV58QmSaYA
AABUNpYJ312ux2REtdygxwzrxEaHYNVaxIwLX7LLB915fYJaQ4C/uuHpx90R8G2n
ph4PwlmHIOTpXPsavpCjmk9qnUMw4SAz1op0NauRv1nT+Mmtx2LD
=Rqyp
-----END PGP MESSAGE-----
```

A questo punto potremmo già usare il file risultante come *reply-block*: quelli a cui lo manderemo dovranno *premetterlo al testo della loro risposta e indirizzarlo al remailer con la cui chiave è stato crittato*, in questo esempio "remailer@anon.efga.org", che provvederà a rinviarcelo senza dunque che essi possano conoscere il nostro vero indirizzo:

```
=================================================================
To: remailer@neva.org
From: joe@freemail.net
Subject: SuperGrulli
-----------------------------------------------------------------
```

```
::
Request-Remailing-To: lex@luthorcorp.com

Caro Lex, come ti dicevo Superman mi sta cercando, e il bottone della
bomba che ho qui con me scotta sempre di piu'.

Ho bisogno di indicazioni operative. Per rispondermi copia il blocco
di testo qua sotto e incollalo all'inizio del messaggio che mi spedi-
rai. Indirizza il messaggio a remailer@anon.efga.org

------ cut here ------ cut here ------ cut here ------ cut here -----

::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
Comment: PGP 2.x compatible

hIwC8cAPVx+T6qkBBACY8/zOp83CZJDeSrb9N1f0hAJLxgcwSjtX6f6Y7dFHUdp8
YpM1I0CSEcZBPwcWlCPMnv9/xkQFhYFLRV9s0ksiw411dPgtKg3YF8h4ZRwdfecp
VcFZD2qqN7UJa/cGp4nASXA/PQmBkgRJcvts3v8Lt4qCaxTFVg0EY4WV58QmSaYA
AABUNpYJ312ux2REtdygxwzrxEaHYNVaxIwLX7LLB915fYJaQ4C/uuHpx90R8G2n
ph4PwlmHIOTpXPsavpCjmk9qnUMw4SAz1opONauRv1nT+Mmtx2LD
=Rqyp
-----END PGP MESSAGE-----
====================================================================
```

Naturalmente, per una sicurezza effettiva, il *reply-block* sarà costituito da una richiesta di rinvio al nostro indirizzo attraverso una catena di remailer e con la procedura di crittazione ormai nota. Ricordiamo che in questo caso il remailer a cui indirizzare il messaggio dovrà essere il primo della catena, cioè l'ultimo nell'ordine dei passaggi di crittazione.

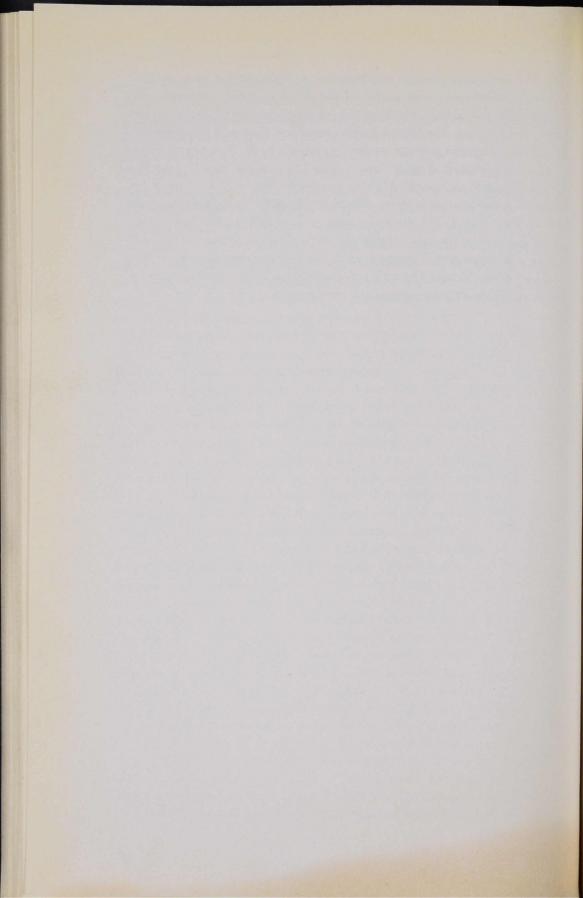
Una volta predisposto, il file potrà essere allegato a tutti i messaggi che desideriamo. Da notare che la catena attraverso cui inviamo la nostra e-mail può benissimo essere del tutto diversa da quella prevista nel *reply block*.

Come si accennava prima, recentemente sono stati implementati dei particolari remailer detti *Nym Server* che riprendono in parte la filosofia del vecchio anon.penet.fi, accoppiando un indirizzo pseudo-anonimo al quale è possibile rispondere direttamente. Ma per questo rimandiamo alla lettura del prossimo capitolo.

Su quel capitolo ci ho lavorato peggio di un negro nei campi di cotone, credimi.

Amico, a Joe Lametta mica gli piace tanto farsi il culo, sai? Se mi piaceva me ne potevo restare nel Bronx, un posto da lavamacchine l'assistente sociale me lo rimediava. Già, una brava donna era. Un po' vecchiotta per un ragazzino come me, e per nulla furba, ma aveva due poppe belle grosse. E nemmeno cadute troppo vicino all'ombelico, ancora. Il posto me lo aveva rimediato per davvero. Per darmi un'opportunità, diceva lei. E anche in ricordo di tutte le volte che su quelle poppe mi aveva lasciato esercitare, credo. Ci restai in tutto due giorni, a lavare quelle macchine, e lei se la prese con me quando mollai. "Irrimediabilmente pigro e propenso a delinquere", disse che ero. Bah. Sul propenso a delinquere nulla da ridire. Ma pigro non lo sono mica. Per come la vedo io, pigri davvero sono quelli che si accontentano di rompersi il culo tutto il giorno per 10 schifosi dollari, e di scappellarsi quando qualche fetente gli lascia la mancia. "Bravo ragazzo, eccoti 10 cents. Però pulisci meglio quell'angolino, c'è rimasta una macchia". "Sì signore". E giù un gran sorriso mangiamerda. E vale anche per quei fetenti di colletti bianchi che lasciano le pidocchiose mance. Si credono di essere chissacchì, ma sono solo dei poveri schiavi come gli altri, solo vestiti un tantino meglio... Ehi, hai ragione, amico. Sto cominciando a fare il filosofo, e non è che ci riesco mica tanto bene, io. Ho perso il filo. Tutta colpa di questo piscio con un'etichetta di whisky incollata sopra. Facciamocene un altro goccio, dai. Dopo la terza bottiglia comincia a sembrarti buono, credi a me. Insomma, era per dirti che quel capitolo su quei gran bastardi di anonymous remailer me lo sono divorato d'un fiato. E ho fatto bene. La connessione di Lex era sotto controllo e se ci scrivevamo direttamente Superman poteva arrivare subito anche alla mia. Invece mi sono messo a usare una nuova connessione e un nuovo indirizzo. Sì, di quelli offerti gratuitamente da uno di quei server che si fanno pubblicità in rete. Ho scoperto che indirizzi di questo tipo se ne trovano a migliaia e, per quel che ne poteva sapere Superman, ognuno di quelli poteva essere il mio. Due ore dopo avevo già informato Luthor di tutto attraverso una catena di remailer. Chi controllava la sua posta vedeva solo un certo numero di

messaggi crittati provenire dai remailer, ma non poteva vedere né da dove, né cosa gli scrivevo. Lex ha risposto immediatamente attraverso un'altra catena. Ci ordinato di schiaffare Lois e Jimmy nel portabagagli della loro macchina e riportarli a Metropolis per incatenarli alla Bomba. Giusto per mettere il Sindaco un altro po' sotto pressione. E di spedire le mutandine di Lois al Daily Planet, a Perry White. Quel porco di un direttore di giornale non avrebbe mica pubblicato nulla lo stesso, lo sapevo bene anch'io, stampa-libera-e-indipendente-del-cazzo. Ma così almeno Superman poteva riconoscerle al SuperOdorato e capire che non scherzavamo mica. Accidenti amico, quante me ne ha dette quella Lois Lane quando se n'è andata senza mutande. Quasi mi dispiaceva vederla portar via da Al e Louie, anche se finalmente un po' di silenzio era un bel sollievo. A me le pupe di carattere sono sempre piaciute. Ma che cazzo, dopo fatto il colpo, con la mia parte potevo comprarmene un centinaio di pupe come quella, e anche meglio.... Eggià, amico. Bravo. Vedo che ci sei arrivato anche tu. QUANTO sarebbe stata la mia parte? Il vecchio Lex si era tenuto nel vago, lo fa sempre. E poi c'era una cosa che mi preoccupava. Avevamo fatto una gran cazzata ad uscire. Lo sapevo io, lo sapeva Lex e lo sapevano persino Al e Louie. Loro erano sollevati perchè Lex era stato gentile con noi, e aveva lasciato perdere qualunque accenno al suo calzolaio di cemento, anche se ora in mezzo a quel casino pure loro avevano finito per sapere tutto, e questo non rientrava nei piani. Ma io lo conosco troppo bene il vecchio Lex. Non scorda mai nulla, non perdona mai nulla, e non è mai tanto pericoloso come quando fa il buon vecchio Babbo Natale. Per cui avevo una bella rogna da grattarmi, capisci? E ne avevo un'altra ancora più immediata. Vedi, questi fottuti remailer sono una gran bella figata, basta che uno solo di quelli che adoperi sia sicuro e stai a posto, capisci? Superman poteva anche fare il giro del mondo a velocità ultraluce per cercare di metterli tutti sotto controllo, ma il nostro omino di ferro se ne è fatti di nemici in questi anni, gente che ha preferito spostare i propri affari il più lontano possibile dal SuperGuastafeste. E pensa che molti di questi in realtà non avevano proprio niente da temere dagli sbirri, erano dei poveracci che al solo pensare di fare qualcosa di illegale se la facevano sotto dall'emozione. Solo che Superman, lo sai com'è fatto. Ogni tanto si rende conto che in realtà a Metropolis la gente comincia a essere un po' stufa di lui. Capirai. Con tutti gli inviti a pranzo a sbafo, e con quello che mangia... per non parlare di tutti quei muri distrutti ogni volta che entra da qualche parte e si dimentica di aprire la porta. Eh, sì. Cercano di farglielo capire con delicatezza, perché non si sa mai quello che può succedere con uno come lui se si incazza. Ma ogni tanto se ne accorge di quello che pensa la gente, e allora per cercare di farsi perdonare cosa fa? Acchiappa il primo disgraziato che gli capita a tiro. Magari qualcuno di quelli che invece di stare col naso per aria quando lui sfreccia su Metropolis si fanno gli affari loro. Già. Fa solo finta di non vederli, ma in realtà ne soffre molto. Ne prende uno a caso e lo porta al commissariato. "È un criminale. L'ho visto con i miei occhi mentre rubava le offerte in chiesa." E chi ha interesse a contraddirlo? Qualche mese di galera, un po' di titoli sui giornali e poi al processo si vedrà. Così sono contenti tutti: sbirri, giornalisti e anche il nostro SuperImbecille che si sente di nuovo utile. Certo, quando poi il tizio esce finalmente di galera, dopo qualche mese o qualche anno di superbestemmie, il suo primo

pensiero è quello di filarsela il più lontano possibile da Metropolis. Beh, amico, mica tutti possono essere come me o come Luthor. A dare del filo da torcere a Superman siamo proprio pochi, puoi giurarlo. Ma ad avere dei conti in sospeso con lui, quella è un'altra storia. Come ti dicevo, se ne è fatti di nemici in questi anni. E alcuni di loro, in giro per il mondo, hanno aperto e gestiscono qualche anonymous remailer. E non hanno nessuna intenzione di fare favori all'omino d'acciaio, capisci? Ne basta uno sicuro. Uno solo. E ne stanno spuntando come funghi. Ma c'era un altro problema. Che Superman si fosse messo a usare Internet anche lui era abbastanza prevedibile. Nessun particolare rischio per me, finora. Lex Luthor sembrava ci avesse parato le palle su tutti i fronti, con tutti questi trucchetti su crittografia e anonimato. Solo che ora, con Superman tra i piedi anche in rete e incazzato nero, le comunicazioni tra me e Luthor dovevano farsi più fitte e frequenti. Botta e risposta. Con 'sti fottuti remailer era un po' scomodo. Ma qualcosa mi diceva che il vecchio Luthor aveva pensato anche a questo...

Nym Server

di Putro

Un *nym server* è una macchina che permette di risolvere il principale inconveniente degli anonymous remailer, ovvero consente scambi bidirezionali di posta mantenendo la garanzia di un totale anonimato. Come già si è visto nel capitolo precedente, i remailer eliminano ogni traccia del mittente di un messaggio: una precauzione necessaria per assicurare l'anonimato, che si paga tuttavia con la difficoltà di ricevere messaggi di risposta. La comunicazione anonima può avvenire, ma solo "a senso unico". I nym server sono nati per ovviare a questo problema, sfruttando parte delle capacità già insite negli anonymous remailer.

In realtà è possibile comunicare in due direzioni anonimamente anche usando dei semplici remailer, ma un simile metodo presuppone che il destinatario legga ed esegua una serie di istruzioni ogni volta che vuole rispondere. Come descritto nel capitolo precedente, si puo usare la tecnica dei reply-block per farsi rispondere: il destinatario dovrà avere la pazienza (e la capacità) di leggere le istruzioni contenute nel messaggio, copiare tutto il reply-block, inserirlo all'inizio del messaggio di risposta e inviare tutto al remailer indicato. Tutte queste procedure devono essere eseguite a mano e sebbene non siano complicate, può capitare di dover corrispondere con una persona che non è in grado di portarle a termine correttamente. Ci sono inoltre alcune limitazioni: ad esempio non è possibile iscriversi a una mailing list gestita in automatico da un software come *majordomo*, cosa che con un nym server è invece possibile fare.

Se questi sono i vantaggi, qual è allora il primo passo da fare per utilizzare un nym server? In primo luogo occorre crearsi uno pseudonimo, detto più comunemente *nym* (un'abbreviazione di *pseudonym*) sul nym server stesso. Da

questo momento in poi il nostro nym ci rappresenterà in toto: sarà la nostra identità fittizia.

Cosa serve per usare un nym?

Basta ben poco per crearsi un nym e per cominciare a usarlo: bisogna avere il PGP versione 2.6 o superiore (al momento la 2.6.3i è probabilmente la migliore versione in circolazione).

Bisogna creare una chiave PGP appositamente per il nym, che non sia cioè la propria chiave PGP personale che si usa normalmente, ma una nuova chiave che si userà solo per la posta *da e per* il nym. Ci si può anche creare più di un nym, in questo caso ogni nym dovrà essere accompagnato dalla sua chiave.

È necessaria la chiave pubblica del (o dei) nym server che si intendono utilizzare, nonché quella di tutti i remailer attraverso cui si faranno circolare i messaggi. Bisognerà usare i remailer di tipo I (cypherpunk) che supportino determinate funzioni ("ek", "pgp", ossia i remailer che supportano in pieno le funzioni del PGP). Queste funzioni sono indicate nella lista di Raph di cui si è parlato nel capitolo sugli anonymous remailer. In particolare va ricordato qui che la funzione "pgp" si riferisce al fatto che il remailer supporta la criptazione con PGP, mentre "ek" riguarda la possibilità di usare il comando <Encrypt-Key> per criptare in modo convenzionale la posta.

Serve naturalmente una normale casella di posta elettronica dove poter ricevere e mandare messaggi crittati col PGP.

Serve un editor che permetta di salvare i messaggi come testo in puro ASCII, senza caratteri di controllo speciali. I software come Eudora o Pegasus usano questo tipo di editor, Word per Windows invece non va bene perché per default salva nel suo formato che non è puro ASCII. Nel dubbio è meglio affidarsi all'editor del DOS o al Notepad di Windows o a qualsiasi altro editor di testo il più scarno possibile.

È necessario creare un reply-block, ossia un pacchetto di istruzioni che permetta al nym server di mandare i messaggi ricevuti dal nostro nym sulla nostra normale casella e-mail.

Bisognerebbe infine fare qualche prova per testare la concatenazione dei remailer da noi scelta, per verificare se in linea di massima la catena funziona regolarmente.

Quanti tipi di nym server esistono?

Molti ricorderanno anon.penet.fi come il primo servizio che ha consentito un certo grado di privacy in rete. Anon.penet.fi era un sistema di account anonimi a cavallo fra l'anonymous remailer e il nym server: permetteva di scrivere messaggi anonimi ma anche di ricevere eventuali risposte. Questo traffico di messaggi in ingresso e in uscita, che chiameremo per comodità "scambio bidirezionale", funzionava solo in virtù dell'esistenza sul server di un database di corrispondenze fra gli indirizzi reali di chi accedeva al servizio e i corrispettivi account anonimi rilasciati. Ad ogni messaggio di risposta, il database verificava quale fosse l'indirizzo reale a cui andava inoltrato: era così possibile ricevere le risposte, ma a scapito della totale inaffidabilità della privacy offerta dal sistema.

L'amministratore di sistema di anon.penet.fi, anche se in buona fede, era infatti in grado di risalire attraverso il database all'indirizzo reale di chiunque avesse chiesto un account sul suo sistema. In una particolare situazione (raccontata altrove in questo libro), sotto pressione dell'Interpol, il sysadmin di anon.penet.fi scelse di rivelare, alle autorità che lo richiedevano, l'identità di almeno un utente del suo sistema. In seguito, un po' per questa storia, un po' per altre polemiche che infierirono su di lui, l'amministratore di sistema decise di chiudere anon.penet.fi, che tuttavia rimane nella memoria come uno dei primi sistemi nati per offrire gratuitamente privacy e anonimato agli utenti della rete.

I nym server permettono, esattamente come faceva anon.penet.fi, di ricevere risposte ai propri messaggi anonimi, tuttavia il loro funzionamento è molto più sicuro del rozzo metodo di mantenere le corrispondenze scritte fra indirizzo anonimo e indirizzo reale. Utilizzando un nym server c'è la totale garanzia che perfino l'amministratore di sistema, anche volendo, non sia in grado di risalire all'identità dei suoi utenti. Come è possibile? Per capire il funzionamento di queste strane macchine, occorre addentrarci un poco di più negli aspetti teorici che li guidano.

Come funziona un nym server?

Esistono al momento due tipi di nym server: il *tipo I*, rappresentato ad esempio da alpha.c2.org (attualmente fermo per problemi di abuso) e il *tipo II*, che è l'evoluzione del primo, rappresentato da nym.alias.net. Nella lista di

Raph i nym server del primo tipo sono indicati come "alpha", quelli del secondo tipo come "newnym".

I nym server sfruttano in maniera combinata le potenzialità del PGP e quelle degli anonymous remailer. Una certa dimestichezza con questi due mezzi è necessaria per poterne comprendere appieno il funzionamento e per poterli utilizzare, limitando al massimo il rischio di un errore umano che ne infici la sicurezza e la validità.

In pratica un nym server non è altro che una macchina presso la quale un utente può registrare un proprio alias (o nym o pseudonimo), vale a dire un nome fittizio che viene associato ad una casella di posta elettronica. L'aspetto innovativo sta nel fatto che né il nym né la casella postale sono direttamente riconducibili all'utente stesso. Ciò è possibile perché tutti i messaggi che transitano in entrata e in uscita da e per il nym server, passano prima attraverso una serie concatenata di anonymous remailer.

Detto così pare poco diverso (e poco più credibile) del servizio offerto da anon.penet.fi. Eppure, con un po' di pazienza (e carta e penna alla mano, se si vuole arrivare a capire fino in fondo) si può arrivare a comprendere la straordinaria e ferrea sicurezza che l'uso combinato di nym server, anonymous remailer e PGP possono offrire.

Supponiamo di scegliere "Smith" come pseudonimo presso il server nym.alias.net: in questo modo si chiede di creare un indirizzo (le tecniche per la creazione del nym saranno illustrate più avanti) che suonerà come "smith@nym.alias.net". Questa è una specie di casella di posta elettronica; tuttavia la posta *non* risiede sul server nym.alias.net e quindi non c'è bisogno di collegarsi direttamente per prelevarla (cosa che metterebbe a rischio l'anonimato). È il server che provvede a spedirla al proprio indirizzo e-mail (quello che si usa normalmente tutti i giorni) tutelando però il nostro anonimato con l'uso di una catena di anonymous remailer.

Supponiamo che tu, Joe Lametta, ricercato dalla polizia e complice di Lex Luthor, abbia bisogno di spedire un messaggio anonimo a Luthor e attenda da lui una risposta.

Decidi di usare un nym server, nello specifico nym.alias.net, su cui preventivamente vai a costruirti un nym che chiami Smith, al quale corrisponde l'indirizzo smith@nym.alias.net. Anche la costruzione del nym va problematizzata, ma di questo parleremo più tardi. Ora cominciamo col risolvere uno fra i molti problemi: come fai tu – Joe Lametta - a ricevere i messaggi di Lex Luthor attraverso nym.alias.net in modo che nessuno, dei

singoli operatori coinvolti, possa in alcun modo risalire a te? Occorre far entrare in gioco i concetti di *catena di anonymous remailer* e di *reply-block*, cioè del pacchetto di istruzioni creato per specificare la serie di anonymous remailer attraverso la quale vanno instradati i messaggi che si attendono. Il *reply-block* è paragonabile a un insieme concentrico di istruzioni, che possono essere lette solo una per volta e solo da uno specifico remailer alla volta perché crittate con la sua chiave pubblica. Si può indicare il proprio reply-block una volta per tutte mentre si crea il nym sul nym server, oppure si può cambiarlo di volta in volta per ogni messaggio: la prima pratica è la più diffusa, ma bisognerà almeno controllare che i remailer attraverso cui si vuole far transitare il proprio messaggio siano sempre attivi.

Dunque, tu – Joe Lametta - scrivi attraverso il tuo nym Smith un messaggio a Lex Luthor; lo invii a nym.alias.net attraverso una catena di remailer, per l'occasione vi integri un un *reply-block* nel quale indichi quali sono i remailer concatenati che vuoi che siano utilizzati per inviarti le eventuali risposte. Tutto questo deve avvenire senza che si possa mai risalire nè alla tua identità, nè alla casella di posta elettronica in cui abitualmente ricevi i messaggi. Conoscendo quest'ultima, infatti, è teoricamente possibile effettuare una serie di controlli incrociati che possono condurre fino alla linea telefonica (e dunque al luogo fisico) da cui ti colleghi in rete. Vediamo come è possibile questo meccanismo così apparentemente contraddittorio:

Tu, Joe Lametta, prepari un messaggio per Lex Luthor che firmi e critti con la chiave di smith@nym.alias.net. Chiudi questo messaggio, come nelle scatole cinesi, in un altro messaggio, che invii a nym.alias.net e critti con la chiave pubblica del server. Chiudi questo ulteriore messaggio in un altro messaggio ancora, che critti con la chiave pubblica dell'ultimo anonymous remailer che decidi di utilizzare. Ripeti l'ultima operazione almeno altre due volte con altri due diversi anonymous remailer. In questa maniera ottieni un messaggio che si può considerare una "cipolla" di PGP.

Ogni "strato" del messaggio può essere letto solo da chi ha la corrispondente chiave PGP: così il primo remailer riceve un messaggio di cui sa solo che proviene da te (Joe Lametta) e che è indirizzato ad un altro remailer. Il secondo remailer sa solo che un messaggio che viene da Remailer 1 va spedito a Remailer 3. Il Remailer 3 sa solo che un messaggio proveniente da Remailer 2 va spedito a nym.alias.net. Nym alias.net riceve un messaggio da Remailer 3 in cui ci sono le istruzioni per inviare un messaggio dal nym Smith a un tale Lex Luthor. Di te, Joe Lametta, si sono perse le tracce al secondo remailer e chi si è già letto i debiti capitoli sa che ora hai una ragionevole sicurezza che nessuno possa in alcun modo risalire a te.

A questo punto occorre risolvere un secondo problema: come fa Lex Luthor a risponderti? Lex Luthor, semplicemente, si vede arrivare un messaggio da smith@nym.alias.net, e risponde a questo indirizzo. Il messaggio arriva al server nym.alias.net. Sul server, in corrispondenza dell'account Smith è conservato il *reply-block* che hai a suo tempo scelto e inviato. Illeggibile a chiunque, il reply-block istruisce solo nym.alias.net a rispedire tutti i messaggi che arrivano per smith@nym.alias.net a Remailer 3. Da qui ricomincia, a ritroso, la catena dei remailer (che può essere la stessa dell'andata ma può anche essere diversa). Remailer 3 riceve da nym.alias net un messaggio di cui sa solo da dove arriva e a chi va: Remailer 2. Remailer 2 riceve un messaggio da Remailer 3 di cui sa solo che deve spedirlo a Remailer 1. Remailer 1 provvederà a spedire a te, Joe Lametta, il messaggio che proviene originariamente da Lex Luthor. Da tutta questa catena, qualsiasi sbirro malizioso che controlli la posta di Luthor e voglia risalire ai suoi interlocutori si perderebbe tra i remailer.

Se si è già capito il funzionamento dei remailer concatenati risulterà ovvio che né nym.alias.net né i singoli remailer usati saranno mai in grado di stabilire una connessione tra la tua casella reale e la tua casella nym (sempre che per pigrizia non si usi un solo remailer, cosa caldamente sconsigliata, ma in questo caso si ricade nella categoria degli errori umani).

In particolare, nym.alias.net saprà solo che i messaggi che riceve Smith vengono mandati al Remailer 3 e non ha modo di sapere cosa avverrà poi perché può leggere solo il primo anello delle istruzioni contenute nel reply-block, in quanto quelle successive sono crittate con le chiavi degli anonymous remailer successivi. La stessa cosa accade agli altri remailer, 3 sa che il messaggio arriva da Smith e che deve essere mandato al Remailer 2, il quale sa che arriva dal Remailer 3 (ma non sa che arriva da Smith) e che deve essere mandato al Remailer 1, e via di questo passo, fino all'ultimo remailer della catena, che sa che il messaggio arriva dal penultimo remailer e che deve essere spedito all'e-mail joe@freemail.net.

Ovviamente la catena dei remailer può essere più o meno lunga, sta a noi decidere quanto, ma sarà opportuno tenere sempre presente alcune considerazioni di base. La prima ci dice che più la catena è lunga, più probabilità ci sono che uno dei tanti remailer sia down o malfunzionante e quindi i messaggi vadano persi. La seconda che una catena troppo corta (ad esempio due o addirittura un solo remailer) è certamente meno sicura.

Costruire il proprio reply-block

Si sarà ormai intuito che il reply-block è la parte fondamentale di tutto il processo. Essendo le informazioni crittate in modo concentrico bisogna partire dall'ultima per risalire via via verso quella più esterna. La scelta dei remailer che si vogliono utilizzare è importante: per una attenta selezione sarà opportuno consultare periodicamente la lista di Raph disponibile in rete, per scegliere solo quei remailer che supportano le funzioni "cpunk", "pgp" e "ek".

Per meglio addentrarci nel mondo concentrico dei reply-block, ecco un esempio pratico di come crearne uno. Supponiamo di voler usare la sequenza:

```
nym (smith@nym.alias.net) -- Remailer AAA -- Remailer BBB --
Remailer CCC -- joe@freemail.net
```

questo sarà il percorso che compirà ogni messaggio ricevuto da smith@nym.alias.net. Supponiamo che ogni remailer critti in modo convenzionale il messaggio con una chiave di nostra scelta: per creare il reply-block si dovrà procedere per passi successivi partendo dall'ultima istruzione che ci interessa, in questo modo:

```
::
Anon-To: joe@freemail.net
Latent-Time: +0:00
Encrypt-Key: password_a
```

Il comando <Latent-Time> dovrebbe essere già familiare perché se ne è parlato nel capitolo sugli anonymous remailer: indica ai remailer di far passare del tempo tra quando ricevono il messaggio e quando lo rispediscono (opzione necessaria per non poter tracciare univocamente i messaggi in uscita e in entrata).

Il comando <Encrypt-key> serve invece per far sì che il remailer critti il messaggio in modo convenzionale con una chiave che viene scelta a proprio piacimento, in questo caso "password_a" (viene crittato il messaggio del mittente, non il reply block).

Dopo aver salvato le prime quattro righe nel file block1.txt, si dovrà crittare questo file con la chiave pubblica del Remailer CCC (l'ultimo della catena) usando il comando:

```
pgp -eat block1.txt CCC@remailer.ccc.com
```

in questo modo si otterrà un file block1.asc di questo tipo:

```
-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
hIwC/nqSW1QDQfUBBACknZMV93wFS2CHOorlgslmEm+alhjIleKwbbTTmeRWC5Rg
/S3vZw+95ZuCZfqxKEOXrgZXz0EwfoyBcpVvf9Pb9D19TqEMTmmL/Jpl1xcxmbJ2
=Bla3
-----END PGP MESSAGE-----
```

Ora bisogna aggiungere il secondo anello di istruzioni: per prima cosa si inserisce l'header

```
::
Encrypted: PGP
```

lasciando una riga vuota tra questo e l'inizio del messaggio crittato, e poi si aggiungono le istruzioni che leggerà il Remailer BBB in questo modo:

```
::
Anon-To: BBB@remailer.bbb.com
Latent-Time: +2:00
Encrypt-Key: password_b

::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i
hIwC/nqSW1QDQfUBBACknZMV93wFS2CHOorlgslmEm+alhjIleKwbbTTmeRWC5Rg
vRoiG8ZhXs4r3E8liFsNtMMf6CUAsdV2ZoX1Hw==
=Bla3
-----END PGP MESSAGE-----
```

La sintassi nella costruzione di un reply-block è molto importante. Ad esempio è fondamentale la presenza di una riga vuota tra il comando <Encrypt-Key> e la riga contente i doppi due punti <::>. Tutto ciò (compresa la prima riga con i <::>) va salvato col nome block2.txt e crittato con la chiave pubblica del Remailer BBB.

```
pgp -eat block2.txt BBB@remailer.bbb.com
```

Si ottiene così un messaggio crittato che solo questo remailer potrà leggere. Metto quindi le istruzioni per il Remailer AAA e critto il tutto con la sua chiave pubblica. Infine aggiungo le istruzioni per nym.alias.net, per cui mi troverò con un messaggio del tipo:

```
::
Anon-To: AAA@remailer.aaa.com
Latent-Time: +1:20
```

```
Encrypt-Key: password_d

::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i

/S3vZw+95ZuCZfqxKEOXrgZXz0EwfoyBcpVvf9Pb9D19TqEMTmmL/Jpl1xcxmbJ2
dHNr1NA6WWaIfVOpR+sluNWFxNYuTkO0Fgtg8cOABRGOKzxjb25maWdAbnltLmFs
cPY/ytBRyZPugrONpLgj0+q6mEjCcgQrxpYQ+1PvFPdDx1GmJ5ogZqW+AVHsNqAp
vRoiG8ZhXs4r3E8liFsNtMMf6CUAsdV2ZoX1Hw==
=Bla3
-----END PGP MESSAGE-----

**
```

Alla fine del reply-block occorre assolutamente ricordarsi di lasciare una riga vuota, seguita da una riga che contiene semplicemente due asterischi. Questo perché il comando <Encrypt-Key> fa sì che qualsiasi cosa sia presente al di sotto dei due asterischi venga crittato. Se non si includono i due asterischi, i remailer sbaglieranno a crittare la nostra posta o addirittura la scarteranno. La creazione del reply-block ricalca quindi il processo utilizzato per spedire posta attraverso i remailer concatenati (e infatti è solo questa la sua funzione: indicare quali remailer usare).

Ricostruiamo adesso quello che succederebbe a un messaggio ricevuto da smith@nym.alias.net: il nym server riceve il messaggio, possiede un reply-block (come quello sopra scritto, fornitogli da Smith) e la chiave pubblica di Smith. Per prima cosa critta il messaggio con la chiave pubblica del nym (Smith), controlla le righe del reply-block che può leggere, vede il comando <Encrypt-Key: password_d> e critta ulteriormente il messaggio con la chiave password_d, mette la parte crittata del reply-block all'inizio del messaggio e poi lo spedisce a Remailer AAA. Il remailer riceve un messaggio di cui può decrittare solo una parte, e qui trova le sue istruzioni, tra cui ci sarà un eventuale e consigliato <Latent-Time>, una eventuale ulteriore crittazione convenzionale del messaggio con la chiave password_c (già crittato con la nostra chiave pubblica e con la chiave password_d) e infine l'indirizzo a cui spedire il messaggio (in questo caso il Remailer BBB). Il Remailer BBB riceve un messaggio di cui può decrittare solo una parte che contiene le istruzioni per lui, in cui c'è scritto di mandare il messaggio al Remailer CCC. Il Remailer CCC riceverà un messaggio in cui potrà decrittare soltanto l'ultimo anello del reply-block; vedrà le sue istruzioni, critterà il messaggio con la chiave password_a e infine lo manderà all'indirizzo joe@freemail.net

Joe Lametta riceverà questo messaggio che dovrà decrittare prima con la chiave password_a, poi con password_b, poi password_c, poi password_d e infine con la chiave privata del suo nym (Smith). È ovvio che tutte le crittazioni convenzionali intermedie non sono obbligatorie e diventano decisamente scomode quando sono troppe. Non bisogna però mai dimenticare che sono una ulteriore garanzia di privacy.

Se tutto quello che abbiamo detto finora sembra decisamente ostico, ci si può consolare sapendo che una volta capito il motivo di una singola operazione si capisce automaticamente il funzionamento complessivo. Se poi proprio non si riesce neanche ad afferrare l'idea di un singolo pezzo dell'intero procedimento, è utile sapere che il processo di creazione di un reply-block può essere facilitato con l'utilizzo di un software apposito come *Private Idaho*. Ma torniamo ai nostri problemi. Una volta creato il reply-block si può pensare alla creazione del nym, ma poiché noi sappiamo che il punto debole di tutto questo sistema è spesso l'inaffidabilità dei remailer, prima di passare a spiegare come si costruisce il proprio nym, ci complicheremo ancora un attimo la vita illustrando una ulteriore comoda funzione del nym server: la possibilità di utilizzare più di un reply-block.

Reply block multipli

Si può utilizzare più di un reply block, specificando ognuno di questi in un messaggio per config@nym.alias.net, l'indirizzo al quale vanno spediti i messaggi di creazione e configurazione del nym. Per esempio:

```
Config:
From: Smith
Reply-Block:
::
Anon-To: AAA@remailer.aaa.com
Latent-Time: +0:00
Encrypt-Key: password_a

::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i

/S3vZw+95ZuCZfqxKEOXrgZXz0EwfoyBcpVvf9Pb9D19TqEMTmmL/Jp11xcxmbJ2
vRoiG8ZhXs4r3E81iFsNtMMf6CUAsdV2ZoX1Hw==
```

```
=Bla3
-----END PGP MESSAGE-----

**

Reply-Block:
::
Anon-To: BBB@remailer.bbb.com
Latent-Time: +1:00r
Encrypt-Key: password_b

::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i

hIwC/nqSW1QDQfUBBACknZMV93wFS2CH0orlgslmEm+alhjI1eKwbbTTmeRWC5Rg
vRoiG8ZhXs4r3E8liFsNtMMf6CUAsdV2ZoX1Hw==
=Bla3
-----END PGP MESSAGE-----

**
```

questo messaggio verrà interpretato dal nym server in modo che una copia di ogni messaggio che riceverà il nym Smith verrà mandata al Remailer AAA (e ad ulteriori altri inclusi nel primo reply-block) immediatamente, mentre un'altra copia verrà mandata al Remailer BBB dopo un tempo random compreso tra 0 minuti e 1 ora (a seconda dell'impostazione del comando <Latent-Time>).

L'utilizzo di diverse catene di remailer rende più complicata l'analisi del traffico da/per la propria casella postale. Per questo motivo è possibile assegnare delle probabilità di utilizzo per diversi reply-block in modo da rendere il tutto molto casuale; per fare ciò bisogna aggiungere un comando del tipo "x=probabilità" alla riga contenente il comando <Reply-Block:> dove x è una singola lettera e rappresenta un indicatore che raggruppa dei reply-block, mentre il valore indica la percentuale di probabilità di utilizzo. Un esempio può chiarire il tutto: per semplificare la visualizzazione di questi esempi si indicherà solo la prima parte dei vari reply-block, in realtà tra la riga <Encrypt-Key:> e la riga con i due asterischi ci deve essere - come visto precedentemente - una parte di testo crittata che contiene il resto delle informazioni per il corretto instradamento del messaggio.

```
Reply-Block: a=0.8
::
Anon-To: AAA@remailer.aaa.com
Latent-Time: +0:00
```

```
Encrypt-Key: password_1

**
Reply-Block: b=0.5
::
Anon-To: BBB@remailer.bbb.com
Latent-Time: +1:00r
Encrypt-Key: password_2

**

Reply-Block: b=0.5
::
Anon-To: CCC@remailer.ccc.com
Latent-Time: +1:00r
Encrypt-Key: password_3

**
```

Da notare en passant la solita presenza dei doppi asterischi dopo una riga vuota alla fine di ogni reply block.

Con un sistema di reply-block di questo tipo abbiamo due gruppi di reply-block: il gruppo contrassegnato dalla lettera "a", costituito da un singolo reply-block, e il gruppo contrassegnato dalla lettera "b", formato dai restanti due reply-block. In questo caso c'è una probabilità di 0.8 (cioè dell'80%) che ogni messaggio venga instradato tramite il remailer AAA, alla quale si aggiunge una ulteriore probabilità del 100% (0.5 + 0.5, cioè una certezza) che venga instradato *anche* al remailer BBB o al remailer CCC, che si spartiscono un 50% di possibilità ciascuno.

Perché aggiungere al già complicato funzionamento dei nym server questa assurda somma di probabilità? Perché si rende estremamente più complessa l'analisi del traffico da e per la propria casella postale e perché se una delle catene di remailer non dovesse funzionare, abbiamo una discreta probabilità che funzioni l'altra senza dunque correre il rischio di perdere qualche messaggio. Si sarà dunque sicuri che almeno una copia di ogni messaggio verrà instradata attraverso i remailer BBB e CCC, ma c'è anche un 80% di probabilità che vengano spedite due copie del medesimo messaggio. Nota finale: è comunque sconsigliato superare il 100% di probabilità associato ad un gruppo di reply-block.

Creare un nym

Ci siamo finalmente arrivati. Creare un nym presso il server è un passaggio delicato e fondamentale, per il quale occorre prestare la massima attenzione. Il messaggio di configurazione deve contenere specifiche istruzioni che possono variare da nym server a nym server: vale dunque prima di tutto il consiglio di prelevare tutti gli help possibili dalle diverse macchine per verificare quale sia la corretta procedura per la specifica macchina che si sceglie di usare. Per ricevere l'help da nym.alias.net è sufficiente spedire un messaggio di posta elettronica a help@nym.alias.net, senza bisogno di aggiungere altro nel subject o nel corpo del messaggio.

D'ora in avanti parleremo di come si lavora su un particolare nym server: nym.alias.net, a cui si consiglia di fare comunque una visita in rete. Non ci sono motivi specifici per giustificare questa scelta: al momento ci sono solo due nym server di tipo newnym attivi (e questo è uno dei due); gli altri nati più recentemente sono ancora del tipo vecchio, sulla scia di alpha.org.

Allora: qual è la procedura per creare il proprio nym su nym.alias.net? In primo luogo occorre scegliersi un alias. Joe Lametta ha scelto ad esempio di usare Smith. È buona norma cercare di evitare alias banali perché probabilmente qualcuno li sta già usando. Se si hanno dubbi, si può ricevere una lista degli alias già esistenti presso nym.alias.net mandando una e-mail a list@nym.alias.net. Se si chiede di creare un alias già esistente la richiesta verrà rifiutata.

Prima di tentare di creare un nym è quindi buona norma controllare che l'alias scelto non esista già, così come è buona norma controllare il funzionamento della catena di remailer che si decide di usare come reply-block con qualche messaggio di prova. Il messaggio di creazione di un nym ha questo formato:

```
Config:
From: Smith
Nym-Commands: create +acksend +fingerkey name="Smith un uomo qualun-
que"
Public-Key:
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.3i

cy5taXQuZWRxPokAlQMFEDGf6ClEwbR1YufHOQEBX6OD/jZ5MFRFIFA1VxTPD5Zj
Xw2bvqJqFvlwLD5SSHCVfe/ka6ALuxZGFKD/pHpUAkfv1hWqAYsJpiOcf8HSdi23
bh5dUeLJnHHHDmd9d55MuNYI6WTi+2YoaiJOZT3C7OoOuzVXuELZ+nZwV2Oyxe8y
4M3bOXjt9kq2upbCNuHZmQP+
=jIEc
-----END PGP PUBLIC KEY BLOCK-----
```

```
Reply-Block:
::
Anon-To: aaa@remailer.aaa.com
Latent-Time: +1:20
Encrypt-Key: passphrase_d

::
Encrypted: PGP

-----BEGIN PGP MESSAGE-----
Version: 2.6.3i

hIwC/nqSW1QDQfUBBACknZMV93wFS2CH0orlgslmEm+alhjI1eKwbbTTmeRWC5Rg
0GsHpQ/TxpazBCVhdBmPblj5wWvwfG1+ZKpIkQ5hiLJhryQM/TUDarEscs3zdaYA
AAB5231aMcQ74AKoDZizABMF3Tw+olV4mm4jVo9cMn2B3Rj2XBF14pV9VL3h0ZQB
cPY/ytBRyZPugr0NpLgj0+q6mEjCcgQrxpYQ+1PvFPdDx1GmJ5ogZqW+AVHsNqAp
vRoiG8ZhXs4r3E8liFsNtMMf6CUAsdV2ZoX1Hw==
=Bla3
-----END PGP MESSAGE-----

**
```

all'interno di questo messaggio si può notare che: la prima riga contiene la parola <Config:>; la seconda riga l'indicazione del nym; la terza un insieme di istruzioni tra cui spicca <create> che appunto indica che si vuole creare un nuovo nym di nome Smith; ci sono poi altri comandi come <+acksend> e <+fingerkey> che verranno descritti più avanti; alla quinta riga comincia la chiave pubblica che è stata precedentemente creata apposta per il nym (chiave che avrà come user ID <smith@nym.alias.net>, a meno che non vogliate renderla pubblica, per i motivi che sono indicati più avanti nelle considerazioni sulla sicurezza); per estrarre questa chiave dal proprio keyring bisogna usare il comando: <pgp -kxa>. Al termine della chiave pubblica comincia il reply-block che si è deciso di utilizzare; chiude il messaggio la solita linea contenente due asterischi <**>. È importante rispettare precisamente l'ordine di queste indicazioni, così come è importante ad esempio che ci sia una linea vuota tra la fine del reply-block e la linea finale con i due asterischi. Un banale errore nella composizione di questo messaggio potrebbe causare l'annullamento della richiesta di creazione senza che però venga notificato alcunché (e quindi l'eventuale perdita di tutti i messaggi che si vorranno mandare attraverso quel nym).

Questo messaggio deve essere firmato con la chiave del nym (Smith) e crittato con la chiave pubblica del server nym.alias.net. Se manca la firma o se non è crittato, il messaggio non viene ritenuto valido. Infine, il messaggio va spedito così com'è a config@nym.alias.net. Come si è già visto, è opportuno utilizzare anche per tutte le operazioni di configurazione una catena di remailer, sempre per garantirsi una più inattaccabile privacy (quindi questo

messaggio non va inviato direttamente a config@nym.alias.net, bensì a questo indirizzo tramite una catena di remailer).

Se si è proceduto nella maniera corretta, se l'alias scelto non esiste già e se la richiesta di configurazione è a posto, il nym server provvederà a inviare un messaggio di conferma attraverso il reply-block che è stato specificato. A questo messaggio bisogna rispondere, per verificare che il reply-block funzioni correttamente. Dopodiché il nuovo nym sarà effettivamente attivo. Per modificare qualche parametro di configurazione o il reply-block basterà mandare un ulteriore messaggio a config@nym.alias.net, senza però inserire il comando <create>.

MANDARE POSTA ATTRAVERSO IL NYM

Mentre ormai diamo per assodato che sia chiaro come fa un messaggio indirizzato al nostro nym ad arrivare fino a noi, resta da specificare come si fa a mandare posta dal proprio nym.

Qui le cose si semplificano un po': semplicemente bisogna scrivere un messaggio, crittarlo con la chiave pubblica del server nym.alias.net, firmarlo con la chiave del proprio nym e inviarlo a send@nym.alias.net facendolo prima passare attraverso qualche remailer. Così, per esempio, creo un file di questo tipo:

```
From: Smith
To: lex@luthorcorp.com
Subject: contatto

Il nostro piano sembra funzionare: Superman è in scacco e non potrà
mai risalire dal nym Smith all'indirizzo di Joe Lametta
```

Se si chiama questo file test1.txt e la chiave PGP del proprio nym ha come user ID 'smith@nym.alias.net', bisognerà crittare e firmare il file con questo comando:

```
pgp -seat test1.txt send@nym.alias.net -u 'smith@nym.alias.net'
```

e quindi spedirlo a send@nym.alias.net (sempre facendolo prima passare attraverso qualche remailer). È fondamentale ricordarsi di non mandare mai niente direttamente a nym.alias.net per evitare di evidenziare un collegamento diretto tra il nym e un indirizzo reale: per questo motivo non ci si stancherà di ripetere che bisogna utilizzare i remailer (concatenati). Se tutto funziona bene Lex Luthor riceverà questo messaggio:

```
From: smith@nym.alias.net
Subject: contatto

Il nostro piano sembra funzionare: Superman è in scacco e non potrà
mai risalire dal nym Smith all'indirizzo di Joe Lametta
```

Se si è in fase di *beta testing*, cioè se si stanno solo facendo delle prove, è bene ricordarsi di farle con amici *veramente* fidati, oppure meglio ancora da soli, magari usando un account diverso (è facile procurarsene più d'uno) per non fare confusione.

Se nella fase "creazione del nym e del reply-block" hai specificato il comando <+acksend> (l'abbiamo visto poco sopra) anche tu riceverai un messaggio dal server che ti confermerà che il messaggio è stato instradato correttamente. Attenzione: il remailer interno del server nym.alias.net (quello che il server utilizza per spedire la posta) mantiene una cache dei messaggi e non accetterà lo stesso messaggio due volte, a meno che ogni copia sia firmata separatamente. Quindi, se si utilizza una lunga catena di remailer, conviene sempre spedire più copie di ogni messaggio, per evitare che vada perso causa il non-funzionamento di anche un solo remailer. Se una o più copie riusciranno a passare, solo una verrà poi processata e rispedita dal nym server. D'altro canto questo filtro presenta qualche inconveniente: se ad esempio si firma un messaggio col PGP per usarlo come test e postarlo quindi più volte, funzionerà solo la prima volta.

Le firme PGP dei messaggi vengono inoltre considerate valide solo per una settimana. Quindi, se un messaggio arriva a send@nym.alias.net più di una settimana dopo che è stato firmato, quel messaggio sarà cestinato.

Considerazioni sulla sicurezza

Il sistema dei nym server presenta un inconveniente legato al PGP. Rendendo pubblica la chiave PGP del proprio nym, qualunque messaggio crittato con quella chiave porta con sé lo USER ID di quella chiave, ad esempio "Smith <smith@nym.alias.net>". Quindi, visto che ogni messaggio spedito attraverso il reply-block viene crittato con la tua chiave pubblica, se qualcuno viene in possesso di questo messaggio non potrà decrittarlo ma saprà che è destinato a chi possiede la chiave segreta di "smith@nym.alias.net" (se infatti si prova a decrittare un messaggio crittato con una chiave pubblica il PGP avvisa che serve una certa chiave privata - e indica anche quale).

Parallelamente lo User ID di una chiave segreta è facilmente ottenibile pur non possedendo la password della chiave stessa: se qualcuno entra in qualche modo in possesso della chiave segreta che usi per il nym (perché può mettere le mani sul tuo computer o per altri motivi), può risalire allo User ID anche senza che tu gli fornisca la password. È ovvio che se si trova una chiave con User ID "Smith <smith@nym.alias.net>" sul tuo hard disk la tua privacy sarà abbastanza compromessa! Quantomeno, anche se nessuno sarà ancora in grado di sapere che cosa è scritto nei messaggi a Smith - ci sarà tuttavia una prova sufficientemente univoca che tu hai qualcosa da spartire con Smith. Non è nulla di determinante, ma a volte può risultare spiacevole. Tuttavia si possono trovare molteplici soluzioni a un simile inconveniente.

Anzitutto non è necessario rendere pubblica la chiave PGP del nym. Si può evitare di distribuirla e usare uno User ID "depistante" come "prova prova" e non firmare i messaggi inviati. Può essere una soluzione definitiva, tuttavia può anche trasformarsi in una limitazione, perché in questo modo non è garantito al cento per cento al destinatario che il messaggio provenga effettivamente da te, anche se il server nym.alias.net firma i messaggi con la sua chiave per evitare falsificazioni. Si può allora usare un particolare software come "premail" (per unix) che mantiene la chiave del tuo nym in un keyring separato e crittato. Manda un messaggio a <premail-info@nym.alias.net> per maggiori informazioni sull'uso di premail. Più rudimentalmente si può tenere la chiave segreta su un dischetto crittato con un software come "Secure File System" o altri (vedi il capitolo sui file system crittati). Infine, si può scegliere di far crittare convenzionalmente (cioè non a chiave pubblica) la posta in arrivo da ogni remailer del reply-block. Da un messaggio crittato convenzionalmente non si può infatti risalire al destinatario.

NYM-COMMANDS

Occorre infine gettare uno sguardo ai Nym-Commands, cioè ai comandi per il nym. Con il loro utilizzo si attivano o disattivano particolari funzioni del nym server. Abbiamo visto che alcuni Nym-Commands possono essere specificati nel momento della creazione del nym, ma possono anche essere trasmessi di volta in volta con ogni comando di configurazione; un messaggio di cambio di configurazione va inviato a config@nym.alias.net e sarà simile a quello usato per la creazione del nym, ma senza la public key e senza

il reply-block (a meno che non si voglia sostituire quello/i che state usando), e soprattutto senza il nym-command <create>. Per attivare o disattivare ogni funzione basta accompagnare il nym-command col segno <+> o con il segno <->. Nym.alias.net supporta i seguenti comandi:

ACKSEND: abilita/disabilita un avviso automatico ogni volta che un messaggio viene rispedito con successo attraverso send@nym.alias.net. Questa opzione può essere definita in ogni singolo messaggio. Ogni volta che tu, smith@nym.alias.net, mandi un messaggio a lex@luthorcorp.com, il nym server ti manda un messaggio di conferma, attraverso il reply block, del fatto che il messaggio è stato spedito nel modo corretto (e che quindi Lex Luthor dovrebbe averlo ricevuto). È molto utile, soprattutto quando si è un po' inesperti e si rischia di fare qualche errore; in questo caso la conferma non arriverà e si saprà che il messaggio è andato perso.

SIGNSEND: Abilita/disabilita la firma PGP automatica di ogni messaggio mandato attraverso i remailer. Bisogna premettere che ogni messaggio che mandi a send@nym.alias.net deve essere firmato e crittato con la chiave del tuo nym (sia che tu renda o non renda pubblica la tua chiave) perché send@nym.alias.net rifiuterà sempre un messaggio a meno che non riesca ad estrarne una firma valida (in caso contrario, chiunque potrebbe inviare messaggi dal tuo nym spacciandosi per te).

Questo riguarda però il messaggio globale, ossia l'intestazione che contiene mittente, destinatario e subject oltre al messaggio vero e proprio per il destinatario. Una volta che send@nym.alias.net ha decrittato il messaggio e verificato la tua firma, si trova a dover gestire il messaggio vero e proprio per il destinatario che può essere crittato oppure in chiaro e firmato oppure non firmato (questo dipende da te, da come hai preparato il messaggio).

Il comando <signsend> è utile nel caso in cui si scelga di non firmare i messaggi per il destinatario con la chiave del proprio nym. In questo caso, per garantire che il messaggio provenga dal nym server, se si attiva questo comando il server firmerà il messaggio con la propria chiave e il destinatario potrà verificare questa firma se possiede la chiave pubblica del server. Se disabiliti questa funzione chiunque può falsificare la posta proveniente dal tuo nym facilmente.

Tutto questo nel caso in cui tu non firmi i tuoi messaggi per i destinatari finali. Se però hai deciso di rendere pubblica la public key del tuo nym, vorrai firmare tutti i messaggi in uscita con la chiave pubblica del tuo nym (vale a dire che se <signsend> è attivo verrebbero firmati una seconda volta, cosa abbastanza inutile). Avere una firma del nym.alias.net attorno a un'altra firma può impedire ai lettori di posta di verificare la firma interna (la tua), così

bisognerà scegliere l'opzione <-signsend> se vuoi firmare da te i messaggi. In caso contrario il destinatario riceverà un messaggio che avrà la firma di send@nym.alias.net all'esterno e poi la tua firma, controllabile solo in un secondo momento. La presenza di una eventuale crittazione (convenzionale o meno) per il destinatario non crea problemi, in tutti i casi l'eventuale firma del send@nym.alias.net è l'involucro più esterno del messaggio.

CRYPTRECV: Abilita/disabilita la crittazione automatica dei messaggi ricevuti per il tuo alias con la chiave pubblica del tuo nym. Disabilitare la crittazione a chiave pubblica riduce la privacy perché un eventuale messaggio non crittato dal mittente passerebbe al primo remailer del tuo reply-block in chiaro, ed eventualmente anche a tutti gli altri se non hai specificato l'utilizzo della crittazione convenzionale durante la creazione del reply-block. In pratica, se il nym server riceve un messaggio da Lex Luthor indirizzato a Smith, nel caso in cui <cryptrecv> sia attivo il server critterà questo messaggio (anche nel caso sia già crittato) con la chiave di Smith e poi lo spedirà al primo remailer specificato nel reply block. Anche quando <+cryptrecv> è abilitato, si può ancora usare la crittazione convenzionale per i remailer concatenati per prevenire che la tua posta sia tracciata.

FIXEDSIZE: Quando mandi il Nym-Command <+fixedsize>, tutti i messaggi che riceverai verranno divisi e/o portati alla stessa identica dimensione (circa 10K). Questo riempimento avverrà all'esterno della crittazione a chiave pubblica, e quindi è utile solo se si usa anche la crittazione convenzionale. Se usi la crittazione convenzionale (come dovresti) avrai tutti i messaggi della stessa dimensione e ciò rende significativamente più complicato per chiunque analizzare il traffico della posta del tuo nym.

DISABLE: Uno dei modi più efficaci per attaccare un remailer è quello di ingolfare il sistema con messaggi per una destinazione particolare. Inoltre, dato che il remailer non è in grado di risalire al destinatario finale di un messaggio, è possibile che qualche furbacchione spedisca i messaggi a sé stesso, magari anche usando due reply-block per creare un livello di traffico che aumenti esponenzialmente. Per proteggersi da abusi di questo genere, il nym server disabilita ogni alias che mandi o riceva più di 10Mb di posta in un giorno. Se si raggiungono livelli di traffico di questo genere, la posta verrà reinviata al mittente. È possibile riabilitare un alias mandando un messaggio con il Nym-Command: <-disable> a config@nym.alias.net.

FINGERKEY: Permette di ricevere la chiave pubblica PGP di un certo nym facendo un finger al suo indirizzo e-mail. Lo user ID della chiave deve necessariamente contenere l'indirizzo nym completo racchiuso fra le parentesi angolari "<" e ">" per poterla ricevere tramite finger. Così, una chiave pubblica con uno user ID come "Smith <smith@nym.alias.net>" sarà

ottenibile con un finger a smith@nym.alias.net, ma la chiave con uno user ID come "smith@nym.alias.net" non lo sarà.

NAME="Smith un uomo qualunque": Di solito le linee del campo From: in una e-mail contengono il nome per intero dello user oltre che il suo indirizzo e-mail. Si può specificare un nome che sia stampato in tutti i tuoi messaggi in uscita, come questo:

```
From: Smith un uomo qualunque <smith@nym.alias.net>
```

e si può far sì che il nome per intero appaia quando il tuo nym subisce una richiesta di finger: è sufficiente mandare il corrispondente <Nym-command name=> in un messaggio di configurazione. Si noti che le virgolette sono necessarie anche se il nome non contiene degli spazi. Se il nome stesso per intero contiene delle virgolette, bisogna farle precedere da un carattere backslash, come, ad esempio:

```
Nym-Commands: name="Smith \"un uomo qualunque\" e anonimo"
```

Per eliminare il tuo nome per intero così che la posta in uscita mostri solo l'indirizzo del tuo nym e il finger mostri al posto del nome per intero i caratteri '???', manda il comando <name="">.

CREATE/CREATE?: Uno di questi due comandi deve essere dato quando si crea un nuovo alias. Il comando <create> verrà respinto se esiste già un nym con quel nome. Il comando <create?> creerà un nuovo nym, ma può anche aggiornare un nym esistente se il messaggio di configurazione è firmato con la chiave privata del nym precedente.

DELETE: Questo comando elimina il tuo alias e cancella il tuo reply block. Come descritto sopra, dovresti ricevere un messaggio firmato col PGP che ti dice chiaramente che il tuo alias è stato eliminato. Un messaggio che conferma semplicemente una generica esecuzione della tua richiesta ("successful execution") non indica che il tuo alias è stato cancellato. Nota che il messaggio non sarà crittato col PGP se hai selezionato <-cryptrecv>, ma in quel caso la crittazione convenzionale con l'appropriata chiave dovrebbe fornire qualche garanzia di autenticità.

I valori di default per i Nym-Commands sono: -acksend -signsend +cryptrecv -fixedsize -disable -fingerkey name=""

Conclusioni

Un consiglio sempre valido è quello di girare in rete per cercare novità e informazioni più dettagliate e diversificate possibili. Nel newsgroup "alt.privacy.anon-server" vengono postati periodicamente gli aggiornamenti sui remailer attivi (compresa la lista di Raph), vengono discusse le problematiche relative a questo tipo di servizi e spesso si trovano risposte ai problemi più o meno comuni. Per semplificare l'utilizzo del nym è consigliato il software Private Idaho per gestire la propria posta da/per nym.alias.net (così come per altri nym server, o anche solo per l'utilizzo dei remailer).

Alcuni consigli spicci, per concludere. Se si ha a cuore la segretezza della propria identità, allora un modo sicuro per proteggerla è di puntare tutti i reply-blocks su un newsgroup usenet e usare un news server che non faccia log dei collegamenti. Non basta contare su questo o quel nym server per proteggerla: essa dipende in misura molto maggiore dal numero e dall'integrità dei remailer attraverso cui sono instradate le risposte. Bisogna anche tenere a mente che spesso sulla stessa macchina su cui è installato un nym server è anche installato un remailer: in simili casi usare questo remailer come ultimo anello di una catena di concatenazione non è un'idea furba...

Poiché l'ambiente digitale è il regno in cui Sherlock Holmes si troverebbe più a suo agio, bisogna sempre considerare che qualche traccia della propria attività può comunque restare incollata da qualche parte (e infatti ci resta): il reply block, la PGP key e le informazioni sulla configurazione del nym saranno ovviamente mantenute sul server per il tempo in cui un account è valido. Anche se si decide di cancellare un nym, le informazioni relative a esso possono resistere come copie di backup ancora per molto tempo (e potranno potenzialmente essere recuperate dagli amministratori del sistema o peggio).

Può essere anche utile sapere che molti nym server conservano anche due dati statistici su ogni nym: prima di tutto l'ammontare della posta ricevuta nelle ultime ventiquattr'ore (dato necessario per rilevare gli attacchi di flood e loop indesiderati con crescite esponenziali, come già visto nel comando <-disable>). Inoltre, il server conserva la data dell'ultimo giorno nel quale è stato inviato un messaggio firmato col PGP a <config@nym.alias.net> o a <send@nym.alias.net>. Se passano 120 giorni senza che tu ripeta un messaggio firmato, il nym server in genere cancella il tuo nym (ti arriveranno comunque molti messaggi di avviso prima che ciò avvenga). La cancellazione, anche se volontaria, non è comunque mai sicura al cento per cento (causa copie di backup, vedi sopra).

Per concludere, gli errori umani. In genere sono le piccole imprecisioni, le dimenticanze o anche solo una chiacchierata al telefono le principali cause della perdita della propria privacy. È dunque sempre buona norma impratichirsi in tempo di pace con le procedure che può essere necessario praticare (in fretta) in casi d'emergenza. Solo una buona pratica e un'ottima comprensione di tutti i meccanismi permettono un uso sufficientemente sicuro di strumenti complessi come i nym server; per questo si consiglia anche di non usare software come Private Idaho, almeno inizialmente, perché la notevole semplificazione delle varie operazioni avviene al prezzo di "occultare" alcune fasi di cui è importante avere invece piena consapevolezza e padronanza.

Nymserver, ecco quello che mi serviva, amico! Potevo usare uno di quelli che mi aveva indicato Lex, ma ormai sapevo il fatto mio e qualcosa mi diceva che avrei fatto meglio a fare qualcosa di più.

Mi ero ricordato di quel cane che cammina come un uomo, quello che mi aveva consegnato la valigetta all'inizio di questa storia, ti rammenti? Beh, anche se è uno sballato e continuava a chiamarmi "lama", dio solo sa perché, è uno che di queste cose se ne intende. Gli ho spiegato la situazione e lui ha capito subito. Detto, fatto: si è messo al lavoro su uno dei serveroni che hanno lì dove lavora lui, alla Walt Disney Corporation. Ah, ora mi ricordo, si chiama Pippo, quel tizio. Beh, cos'hai da stupirti? È uno molto famoso dici? Bah, se me lo dici tu può essere, penserai mica che Joe Lametta passi il suo tempo a leggere fumetti, no? Il qui presente ha lasciato perdere tutte quelle stronzate a sette anni, quando ha visto che strano effetto gli faceva Playboy. Come ti dicevo, già che c'era, quel Pippo ha messo su non solo un remailer normale, ma proprio uno di quei cosi speciali, un nymserver. Così io e il vecchio Lex potevamo tenerci in contatto senza troppe menate. Una bella comodità, vero? Come? Se potevo fidarmi di quel Pippo? Beh, amico. Fiducia è una parola grossa, non c'è che dire, e non la uso tanto volentieri, io. Ma quel Pippo è un tizio strano, non è mica come lo vedi nei film. In realtà lui dice che i suoi cartoni rincretiniscono i bambini, pensa un po'. Li fa solo per la grana, nel tempo libero invece si fa come una scimmia. E appena può scappa nel deserto dai suoi amici fricchettoni. Ho idea che lui col vecchio Lex ci vada a letto solo per il gusto di farlo, mi capisci? Non stupisce che si sia infilato dentro queste faccende su crittografia, Internet e via dicendo. Ci trovi un mucchio di svitati lì in mezzo. Gente che si sbatte su queste cose per il piacere di farlo e non ci cava un dollaro. A me sembrano scemi, ma se va bene a loro va bene anche a me. Non è stata fiducia la mia, diciamo che è stato un rischio calcolato. E del resto che avevo fatto bene il calcolo lo si è visto poi. Perché Superman, amico, ormai era incazzato nero. Hai presente un toro se gli cacci in culo un peperoncino messicano a primavera? Ecco, le mutandine di Lois gli avevano fatto quell'effetto. Quando finalmente ci ha pensato un po' sopra con calma si è lanciato in volo come una furia, con i passeri che gli

cascavano a terra incendiati lungo la scia. Dopo cinque minuti aveva già iniziato a mettere sotto controllo tutti i remailer degli Stati Uniti. Ne rimaneva qualche dozzina sparsa per il mondo, Cecenia, Corea, Libia, quei posti lì. Non fosse stato per evitare di creare grattacapi al vecchio zio Sam, il nostro omino d'acciaio era già pronto a sfasciare a manate i missili antiaerei che gli avrebbero tirato contro non appena si fosse affacciato ai cieli di quei posticini. Ma quando ha visto che c'era di mezzo anche il nym di Pippo e che la roba che entrava lì non si sapeva dove usciva poi, c'è rimasto come un idiota. Perché amico, a quel Pippo mica poteva andare troppo a bacargli il cazzo, sai? Punto primo, Superman è sotto contratto con la DC comics, e la Walt Disney la DC se la mangia in un boccone. E alla Walt Disney non gli va tanto bene che si sappiano in giro troppe cose sui suoi eroi, quindi quel nymerserver tornava utile anche a lei. A proposito, ti ricordi quel fumettaro mangiaspaghetti che anni fa si mise a pubblicare delle storiellette sulla vera vita di Pippo? Beh, amico, quel tizio ora non pubblica più nulla. È schiattato secco. Era un tossico pure lui, dicono, e sarà anche vero. Ma il vecchio Walt ai suoi tempi era uno parecchio ammanicato con la CIA, la sapevi questa? Collaborava al controllo dei sovversivi e a tutta quella roba là. E non mi stupirebbe se la Walt Disney avesse ancora qualche buon contatto da usare all'occorrenza, mi spiego? Punto secondo, quel Pippo pare che se butta giù una nocciolina diventa un cattivo cliente pure per Superman. Sta di fatto che il nostro Mascellone d'Acciaio se l'è beccato tra le chiappone ancora una volta. E mentre se ne stava lì a grattarsi il capoccione rimuginando sulla prossima mossa, io mi sono rimesso a studiare. Perché c'erano ancora alcune cose che mi preoccupavano. Un qualunque ladruncolo da strapazzo a quel punto si sarebbe sentito come un topo nel formaggio: nascosto nel mio covo appena fuori dalla portata della bomba piazzata nelle fogne di Metropolis, la dispensa ben fornita di vino italiano. Computer portatile e connessione in rete. Comunicavo con Lex Luthor una decina di volte al giorno. Gli sbirri sapevano dov'era Lex, ma sapevano anche che da lui non avrebbero trovato nulla di utile, e soprattutto sapevano che il bottone per far saltare la bomba ce l'avevo io. Avevano messo sotto controllo la posta che arrivava a Luthor, nel tentativo di rintracciarmi. Noi sapevamo che loro sapevano. Ma tutti facevano finta di niente. Molto divertente, amico, te lo immagini? Ma il qui presente è un professionista serio, mica un ladro di polli. E non ci ho messo molto a capire che tutti questi giochetti, in fondo, si basavano sul PGP, sulle firme digitali e tutta quella roba. Roba un po' troppo appariscente, capisci cosa voglio dire? Un messaggio crittografato balza all'occhio subito perfino all'ultimo sturacessi dell'ultimo comando di polizia di Metropolis. Figuramoci a un SuperSbirro. Non che Superman potesse farci molto, a questo punto, ma non mi sentivo troppo tranquillo…

Steganografia

di Frank Sinapsi

La parola *steganografia* deriva dall'unione dei due vocaboli greci στεγω (rendo occulto, nascondo) e γραφη (la scrittura). Steganografia è dunque "la scrittura nascosta" o meglio ancora l'insieme delle tecniche che consente a due o più persone di comunicare in modo tale da nascondere non tanto il contenuto (come nel caso della crittografia), ma la stessa esistenza della comunicazione agli occhi di un eventuale osservatore, tradizionalmente denominato "nemico". Si tratta di un'idea tutt'altro che nuova e che anzi vanta origini molto antiche. Nel corso dei secoli sono stati escogitati numerosi metodi steganografici, tutti molto diversi tra loro. Al fine di comprendere meglio l'essenza di questa idea, può essere utile accennare ad alcuni esempi tra i più interessanti e ingegnosi.

Erodoto racconta la storia di un nobile persiano che fece tagliare a zero i capelli di uno schiavo fidato al fine di poter tatuare un messaggio sul suo cranio; una volta che i capelli furono ricresciuti, inviò lo schiavo alla sua destinazione, con la sola istruzione di tagliarseli nuovamente.

Un acrostico è una poesia - o un testo di qualsiasi tipo - composta intenzionalmente in modo tale che, unendo le prime lettere di ogni capoverso, si ottiene un messaggio di senso compiuto. Esistono numerose varianti di questa semplice idea di base, come il testo che segue (il quale fu realmente inviato da una spia tedesca durante la seconda guerra mondiale):

```
Apparently neutral's protest is thoroughly discounted and ignored.
Isman hard hit. Blockade issue affects pretext for embargo on by
products, ejecting suets and vegetable oils.
```

Considerando in sequenza la seconda lettera di ogni parola, si ottiene il messaggio:

```
Pershing sails from NY June 1[1]
```

Le griglie di Cardano erano fogli di materiale rigido nei quali venivano ritagliati fori rettangolari a intervalli irregolari; applicando la griglia sopra un foglio di carta bianca, il messaggio segreto veniva scritto nei buchi (ciascun buco poteva contenere una o più lettere), dopodiché si toglieva la griglia e si cercava di completare la scrittura del resto del foglio in modo da ottenere un messaggio di senso compiuto, il quale poi veniva inviato a destinazione. Applicando sul foglio una copia esatta della griglia originaria, era possibile leggere il messaggio nascosto.

Gli inchiostri invisibili (o inchiostri simpatici) sono sostanze che, in condizioni normali, non lasciano tracce visibili se usate per scrivere su un foglio di carta, ma diventano visibili (rivelando in tal modo la scrittura) se il foglio viene sottoposto a una fonte di calore. È così possibile scrivere il messaggio segreto negli spazi compresi tra le righe di un messaggio dall'aspetto innocuo, quest'ultimo scritto con un inchiostro normale. (Per accedere al messaggio segreto occorre letteralmente "saper leggere tra le righe"...). Le sostanze più usate a questo scopo sono molto comuni: succo di limone, aceto, latte, ma durante la seconda guerra mondiale sono state impiegate sostanze più sofisticate, come ad esempio gli inchiostri al cobalto, che possono essere resi visibili solo mediante l'uso di particolari reagenti chimici.

Un'altra tecnica molto usata dai nazisti durante la seconda guerra mondiale è quella dei micropunti fotografici: si tratta di fotografie della dimensione di un punto dattiloscritto che, una volta sviluppate e ingrandite, possono diventare pagine stampate di buona qualità.

Questi pochi ma significativi esempi dovrebbero essere sufficienti a chiarire il concetto di steganografia, il quale viene spesso confuso, in prima analisi, con quello di crittografia. Per rendere più esplicite le differenze tra questi due concetti possiamo osservare che, mentre nel caso della crittografia è consentito al nemico di rilevare, intercettare e modificare i messaggi senza però avere la possibilità di violare le misure di sicurezza garantite dallo specifico sistema crittografico (cioè senza poter accedere all'informazione vera e propria e quindi leggerne il contenuto), l'obiettivo della steganografia è invece quello di nascondere un messaggio *dentro* un altro messaggio, dall'aspetto innocuo, in modo che il nemico non possa neppure rilevare l'esistenza del primo messaggio.

[1] in realtà c'è una "r" di troppo

UNA E MOLTE STEGANOGRAFIE: LA STEGANOGRAFIA SOSTITUTIVA

Ciò che caratterizza la steganografia, come si è visto, è l'esistenza di un secondo messaggio facilmente percepibile, il cui senso è generalmente del tutto disgiunto da quello del messaggio segreto che esso contiene. Nel seguito si indicherà questo secondo messaggio come *messaggio contenitore* o più semplicemente *contenitore*.

Come si può facilmente immaginare, le nuove tecnologie e in particolar modo i sistemi per l'elaborazione dell'informazione, hanno consentito anche nel caso della steganografia la progettazione di nuove tecniche, sempre più sofisticate, sicure e pratiche da usare. Questo capitolo cercherà proprio di esporre le idee che stanno alla base dei più diffusi programmi per computer che consentono di steganografare un messaggio dentro un altro.

Le prime definizioni proposte riguardano l'origine del file contenitore: alcune tecniche - probabilmente le più numerose – come si vedrà consentono di "iniettare" il messaggio segreto dentro un messaggio contenitore già esistente, modificandolo in modo tale sia da contenere il messaggio sia da risultare, al livello al quale viene percepito dai sensi umani, praticamente indistinguibile dall'originale. Indichiamo l'insieme di queste tecniche con il termine *steganografia iniettiva*. Esistono tuttavia altre tecniche steganografiche che hanno capacità proprie di generare potenziali messaggi contenitori e utilizzano il messaggio segreto per "pilotare" il processo di generazione del contenitore. Per queste tecniche adottiamo il termine *steganografia generativa*. Alla fine del capitolo verranno passati in rassegna esempi di programmi di entrambi i tipi.

Secondo un sistema di classificazione diverso, le tecniche steganografiche possono essere ripartite in tre classi: *steganografia sostitutiva*, *steganografia selettiva* e *steganografia costruttiva*.

Le tecniche del primo tipo sono di gran lunga le più diffuse, tanto che in genere con il termine steganografia ci si riferisce implicitamente a esse. Tali tecniche si basano sulla seguente osservazione: la maggior parte dei canali di comunicazione (linee telefoniche, trasmissioni radio, ecc.) trasmettono segnali che sono sempre accompagnati da qualche tipo di rumore. Questo rumore può essere sostituito da un segnale - il messaggio segreto - che è stato trasformato in modo tale che, a meno di conoscere una chiave segreta, è indistinguibile dal rumore vero e proprio, e quindi può essere trasmesso senza destare sospetti.

Quasi tutti i programmi che sono facilmente reperibili si basano su questa idea, sfruttando la grande diffusione di file contenenti una codifica digitale di immagini, animazioni e suoni; spesso questi file sono ottenuti da un processo di conversione analogico/digitale e contengono qualche tipo di rumore. Per esempio, uno scanner può essere visto come uno strumento di misura più o meno preciso. Un'immagine prodotta da uno scanner, da questo punto di vista, è il risultato di una specifica misura e come tale è soggetta a essere affetta da errore.

La tecnica base impiegata dalla maggior parte dei programmi, consiste semplicemente nel sostituire i "bit meno significativi" delle immagini digitalizzate con i bit che costituiscono il file segreto (i bit meno significativi sono assimilabili ai valori meno significativi di una misura, cioè quelli che tendono a essere affetti da errori.) Spesso l'immagine che ne risulta non è distinguibile a occhio nudo da quella originale ed è comunque difficile dire se eventuali perdite di qualità siano dovute alla presenza di informazioni nascoste oppure all'errore causato dall'impiego di uno scanner poco preciso, o ancora alla effettiva qualità dell'immagine originale prima di essere digitalizzata.

Per fissare meglio le idee, ecco un esempio concreto. Uno dei modi in cui viene solitamente rappresentata un'immagine prodotta da uno scanner è la codifica RGB a 24 bit: l'immagine consiste di una matrice MxN di punti colorati (pixel) e ogni punto è rappresentato da 3 byte, che indicano rispettivamente i livelli dei colori primari rosso (Red), verde (Green) e blu (Blue) che costituiscono il colore. In questo modello i colori possibili sono $2^{24} \cong$ 16milioni; i cosiddetti *grigi* sono i colori in cui i livelli di rosso, verde e blu sono coincidenti e quindi sono soltanto 256. Supponiamo che uno specifico pixel di un'immagine prodotta da uno scanner sia rappresentato dalla tripla (12, 241, 19) (si tratta di un colore tendente al verde, dato che la componente verde predomina fortemente sulle altre due); in notazione binaria, le tre componenti sono:

```
12 = 00001100
241 = 11110001
19 = 00010011
```

quelli che in precedenza abbiamo chiamato i "bit meno significativi" dell'immagine sono gli ultimi a destra, cioè 0-1-1, e sono proprio quelli che si utilizzano per nascondere il messaggio segreto. Se volessimo nascondere in quel pixel l'informazione data dalla sequenza binaria 101, allora bisognerebbe effettuare la seguente trasformazione:

```
00001100 --> 00001101 = 13
11110001 --> 11110000 = 240
00010011 --> 00010011 = 19
```

La tripla è così diventata (13, 240, 19); si noti che questo tipo di trasformazione consiste nel sommare 1, sottrarre 1 o lasciare invariato ciascun livello di colore primario, quindi il colore risultante differisce in misura minima da quello originale. Dato che un solo pixel può contenere un'informazione di 3 bit, un'immagine di dimensioni MxN può contenere un messaggio segreto lungo fino a (3*M*N)/8 byte, per esempio un'immagine 1024x768 può contenere 294912 byte.

La tecnica appena descritta rappresenta il cuore della steganografia sostitutiva, anche se di fatto ne esistono numerose variazioni. Innanzitutto è ovvio che tutto quello che abbiamo detto vale non solo per le immagini, ma anche per altri tipi di media, per esempio suoni e animazioni digitalizzati. Inoltre - e questo è meno ovvio - lavorando con le immagini come file contenitori non sempre si inietta l'informazione al livello dei pixel, ma si è costretti a operare su un livello di rappresentazione intermedio; è questo il caso, per esempio, delle immagini in formato JPEG, nel quale le immagini vengono memorizzate solo dopo essere state compresse con una tecnica che tende a preservare le loro caratteristiche visive piuttosto che l'esatta informazione contenuta nella sequenza di pixel. Se iniettassimo delle informazioni in una bitmap e poi la salvassimo in formato JPEG, le informazioni andrebbero perdute, poiché non sarebbe possibile ricostruire la bitmap originale. Per poter utilizzare anche le immagini JPEG come contenitori, è tuttavia possibile iniettare le informazioni nei coefficienti di Fourier ottenuti dalla prima fase di compressione.

Esiste un altro caso interessante che merita di essere discusso, ed è quello dei formati di immagini che fanno uso di *palette*. La palette (tavolozza) è un sottoinsieme prestabilito di colori. Nei formati che ne fanno uso, i pixel della bitmap sono vincolati ad assumere come valore uno dei colori presenti nella palette: in questo modo è possibile rappresentare i pixel con dei puntatori alla palette, invece che con la terna esplicita RGB (red, green and blue). Ciò in genere permette di ottenere dimensioni inferiori della bitmap, ma il reale vantaggio è dato dal fatto che le schede grafiche di alcuni anni fa utilizzavano proprio questa tecnica e quindi non potevano visualizzare direttamente immagini con un numero arbitrario di colori. Il caso più tipico è quello delle immagini in formato GIF con palette di 256 colori, ma le palette possono avere anche altre dimensioni. Come è facile immaginare, un'immagine appena prodotta da uno scanner a colori sarà tipicamente costituita da più di 256 colori diversi, tuttavia esistono algoritmi capaci di ridurre il numero dei colori utilizzati mantenendo il degrado della qualità entro limiti accettabili. Si può osservare che, allo stesso modo in cui avviene con il formato JPEG, non è possibile iniettare informazioni sui pixel prima di convertire l'immagine in

formato GIF, perché durante il processo di conversione c'è perdita di informazione (osserviamo anche che questo non vale per le immagini a livelli di grigi: tali immagini infatti sono particolarmente adatte per usi steganografici.)

La soluzione che viene di solito adottata per usare immagini GIF come contenitori è dunque la seguente: si riduce il numero dei colori utilizzati dall'immagine a un valore inferiore a 256 ma ancora sufficiente a mantenere una certa qualità dell'immagine, dopodiché si finisce di riempire la palette con colori molto simili a quelli rimasti. A questo punto, per ogni pixel dell'immagine, la palette contiene più di un colore che lo possa rappresentare (uno è il colore originale, gli altri sono quelli simili ad esso che sono stati aggiunti in seguito), quindi abbiamo una possibilità di scelta. Tutte le volte che abbiamo una possibilità di scelta fra più alternative, abbiamo la possibilità di nascondere un'informazione: questo è uno dei principi fondamentali della steganografia. Se le alternative sono *due* possiamo nascondere *un bit* (se il bit è 0, scegliamo la prima, se è 1 la seconda); se le alternative sono *quattro* possiamo nascondere *due bit* (00 -> la prima, 01 -> la seconda, 10 -> la terza, 11 -> la quarta) e così via.

La soluzione appena discussa dell'utilizzo di GIF come contenitori è molto ingegnosa ma purtroppo presenta un problema: è facile scrivere un programma che, presa una GIF in ingresso, analizzi i colori utilizzati e scopra le relazioni che esistono tra di essi; se il programma scopre che l'insieme dei colori utilizzati può essere ripartito in sottoinsiemi di colori simili, è molto probabile che la GIF contenga informazione steganografata. Di fatto, questo semplice metodo di attacco è stato portato avanti con pieno successo da diverse persone ai programmi che utilizzano immagini a palette come contenitori, tanto che qualcuno ha finito per sostenere che non è possibile fare steganografia con esse.

Per mostrare quanto sia ampia la gamma di tecniche steganografiche, accenniamo a un'altra possibilità di nascondere informazioni dentro immagini GIF. Come abbiamo detto, in questo formato viene prima memorizzata una palette e quindi la bitmap (compressa con un algoritmo che preserva completamente le informazioni) consistente di una sequenza di puntatori alla palette. Se scambiamo l'ordine di due colori della palette e corrispondentemente tutti i puntatori ad essi, otteniamo un file diverso che corrisponde però alla stessa immagine, dal punto di vista dell'immagine il contenuto informativo dei due file è identico. La rappresentazione di immagini con palette è quindi intrinsecamente ridondante, dato che ci permette di scegliere un qualsiasi ordine dei colori della palette (purché si riordinino

corrispondentemente i puntatori a essi). Se i colori sono 256, esistono 256! modi diversi di scrivere la palette, quindi esistono 256! file diversi che rappresentano la stessa immagine. Inoltre è abbastanza facile trovare un metodo per numerare univocamente tutte le permutazioni di ogni data palette (basta, per esempio, considerare l'ordinamento sulle componenti RGB dei colori). Dato che abbiamo 256! possibilità di scelta, è possibile codificare log(256!) = 1683 bit, cioè 210 byte. Si noti che questo numero è indipendente dalle dimensioni dell'immagine, in altre parole è possibile iniettare 210 bytes anche su piccole immagini del tipo icone 16x16 semplicemente permutando in modo opportuno la palette.

Dopo avere esaminato alcune tecniche steganografiche di tipo sostitutivo, discutiamo adesso i problemi relativi alla loro sicurezza. Innanzitutto premettiamo che le norme che valgono generalmente per i programmi di crittografia dovrebbero essere osservate anche per l'utilizzo dei programmi steganografici. Per ciò che riguarda le specifiche caratteristiche della steganografia, si tengano presente i seguenti principi: in primo luogo si eviti di usare come contenitori file prelevati da siti pubblici o comunque noti (per esempio, immagini incluse in pacchetti software, ecc.); in secondo luogo si eviti di usare più di una volta lo stesso file contenitore (l'ideale sarebbe quello di generarne ogni volta di nuovi, mediante scanner e convertitori da analogico a digitale, e distruggere gli originali dopo averli usati).

Come si è visto, queste tecniche consistono nel sostituire un elemento di scarsa importanza (in certi casi di importanza nulla) da file di vario tipo, con il messaggio segreto che vogliamo nascondere. Quello che viene ritenuto il principale difetto di queste tecniche è che in genere la sostituzione operata può alterare le caratteristiche statistiche del rumore presente nel media utilizzato. Lo scenario è il seguente: si suppone che il nemico disponga di un modello del rumore e che utilizzi tale modello per controllare i file che riesce a intercettare. Se il rumore presente in un file non è conforme al modello, allora il file è da considerarsi sospetto. Si può osservare che questo tipo di attacco non è per niente facile da realizzare, data l'impossibilità pratica di costruire un modello che tenga conto di tutte le possibili sorgenti di errori/ rumori, tuttavia in proposito esistono degli studi che in casi molto specifici hanno avuto qualche successo.

La steganografia selettiva e quella costruttiva hanno proprio lo scopo di eliminare questo difetto della steganografia sostitutiva. Vediamo di cosa si tratta.

Steganografia selettiva

La steganografia selettiva ha valore puramente teorico e, per quanto se ne sappia, non viene realmente utilizzata nella pratica. L'idea su cui si basa è quella di procedere per tentativi, ripetendo una stessa misura fintanto che il risultato non soddisfa una certa condizione. Facciamo un esempio per chiarire meglio. Si fissi una funzione hash semplice da applicare a un'immagine in forma digitale (una funzione hash è una qualsiasi funzione definita in modo da dare risultati ben distribuiti nell'insieme dei valori possibili; tipicamente questo si ottiene decomponendo e mescolando in qualche modo le componenti dell'argomento); per semplificare al massimo, diciamo che la funzione vale 1 se il numero di bit uguali a 1 del file che rappresenta l'immagine è pari, altrimenti vale 0 (si tratta di un esempio poco realistico ma, come dicevamo, questa discussione ha valore esclusivamente teorico). Così, se vogliamo codificare il bit 0 procediamo a generare un'immagine con uno scanner; se il numero di bit dell'immagine uguali a 1 è dispari ripetiamo di nuovo la generazione, e continuiamo così finché non si verifica la condizione opposta. Il punto cruciale è che l'immagine ottenuta con questo metodo contiene effettivamente l'informazione segreta, ma si tratta di un'immagine "naturale", cioè generata dallo scanner senza essere rimanipolata successivamente. L'immagine è semplicemente sopravvissuta a un processo di selezione (da cui il nome della tecnica), quindi non si può dire in alcun modo che le caratteristiche statistiche del rumore presentano una distorsione rispetto a un modello di riferimento. Come è evidente, il problema di questa tecnica è che è troppo dispendiosa rispetto alla scarsa quantità di informazione che è possibile nascondere. Ad ogni modo, alla fine del capitolo si proporrà un esempio di programma che implementa una steganografia di tipo generativo, utilizzando con successo l'idea di base della steganografia selettiva di nascondere le informazioni procedendo per tentativi.

Steganografia costruttiva

La steganografia costruttiva affronta lo stesso problema nel modo più diretto, tentando di sostituire il rumore presente nel medium utilizzato con l'informazione segreta opportunamente modificata in modo da imitare le caratteristiche statistiche del rumore originale. Secondo questa concezione, un buon sistema steganografico dovrebbe basarsi su un modello del rumore e adattare i parametri dei suoi algoritmi di codifica in modo tale che il falso rumore contenente il messaggio segreto sia il più possibile conforme al modello. Questo approccio è senza dubbio valido, ma presenta anche alcuni svantaggi. Innanzitutto non è facile costruire un modello del rumore: la costruzione di un modello del genere richiede grossi sforzi ed è probabile che qualcuno, in grado di disporre di maggior tempo e di risorse migliori, riesca a costruire un modello più accurato, riuscendo ancora a distinguere tra il rumore originale e un sostituto. Inoltre, se il modello del rumore utilizzato dal metodo steganografico dovesse cadere nelle mani del nemico, egli lo potrebbe analizzare per cercarne possibili difetti e quindi utilizzare proprio il modello stesso per controllare che un messaggio sia conforme a esso. Così, il modello, che è parte integrante del sistema steganografico, fornirebbe involontariamente un metodo di attacco particolarmente efficace proprio contro lo stesso sistema.

Cosa fare? Attenersi al principio di Kerckhoff

A causa di questi problemi, la semplice tecnica iniettiva di base rimane quella più conveniente da usare. Se si hanno particolari esigenze di sicurezza, esiste sempre una strategia molto semplice e allo stesso tempo molto efficace: quella che consiste nell'utilizzare contenitori molto più ampi rispetto alla quantità di informazioni da nascondere. Per esempio, invece di utilizzare i bit meno significativi di tutti i pixel di un'immagine, si può giocare sul sicuro utilizzando solo un pixel ogni 10, o anche più, fino a rendere impossibile, a tutti gli effetti pratici, la rilevazione di una distorsione delle caratteristiche statistiche del rumore. Su questo punto si tornerà in seguito.

Resta da affrontare un'ultima questione molto importante. Abbiamo accennato all'eventualità che i dettagli di funzionamento di un sistema steganografico possano cadere nelle mani del nemico. In ambito crittografico si danno le definizioni di vari livelli di robustezza di un sistema, a seconda della

Figura 1

capacità che esso ha di resistere ad attacchi basati su vari tipi di informazioni a proposito del sistema stesso. In particolare, i sistemi più robusti sono quelli che soddisfano i requisiti posti dal *principio di Kerckhoff*, che formulato in ambito steganografico suona più o meno così: la sicurezza del sistema deve basarsi sull'ipotesi che il nemico abbia piena conoscenza dei dettagli di progetto e implementazione del sistema stesso; la sola informazione di cui il nemico non può disporre è una sequenza (corta) di numeri casuali - la chiave segreta - senza la quale, osservando un canale di comunicazione, non deve avere neanche la più piccola possibilità di verificare che è in corso una comunicazione nascosta.

Se si vuole aderire a questo principio, è evidente che le tecniche esposte fin qui non sono ancora soddisfacenti per caratterizzare un sistema steganografico completo. Infatti, se i dettagli di implementazione dell'algoritmo sono resi di dominio pubblico, chiunque è in grado di accedere a eventuali informazioni nascoste, semplicemente applicando il procedimento inverso (nell'esempio visto, ciò si ottiene "riaggregando" i bit meno significativi dell'immagine). Per affrontare questo problema, è necessario introdurre una fase di pre-elaborazione del file segreto, che lo renda non riconoscibile - da parte del nemico - come portatore di informazioni significative. La soluzione più ovvia è quella di impiegare un sistema di crittografia convenzionale (per esempio, il PGP), il quale garantisce appunto l'inaccessibilità da parte del nemico al messaggio vero e proprio. Lo schema che ne risulta è rifortato nella Figura 1.

La storia purtroppo non è finita qui, perché in questo meccanismo a due stadi il secondo processo è reversibile; in altri termini, chiunque può estrarre il file costituito dalle informazioni che fluiscono dal primo al secondo stadio. Poiché si presume che un crittoanalista esperto possa facilmente riconoscere un file prodotto da un programma di crittografia convenzionale, questo schema è ancora da considerarsi incompleto. Questo punto è di importanza fondamentale, perché rende definitivamente non valido il sistema steganografico, indipendentemente dal fatto che il contenuto dell'informazione segreta resti inaccessibile. Mentre il progettista di un algoritmo di crittografia assume che il nemico impiegherà tutte le risorse possibili per decrittare il messaggio, il progettista di un sistema steganografico deve supporre infatti

che il nemico tenterà di rilevare la sola esistenza del messaggio. In altre parole, la crittografia fallisce il suo scopo quando il nemico legge il contenuto del messaggio: la steganografia invece fallisce quando il nemico si rende semplicemente conto che esiste un messaggio segreto dentro il file contenitore, pur non potendolo leggere. È opportuno quindi che il messaggio crittografato, prima di essere immerso nel contenitore, venga "camuffato" in modo da diventare difficilmente distinguibile da semplice rumore.

A questo scopo, sono stati escogitati diversi metodi. Il più semplice è quello di eliminare dal file crittato da PGP tutte le informazioni che lo identificano come tale: il PGP, infatti, genera un file che rispetta un particolare formato, contenente, oltre al blocco di dati cifrati vero e proprio, informazioni piuttosto ridondanti che facilitano la gestione del file da parte dello stesso PGP (o di shell in grado di trattare con questo formato).

Esiste un piccolo programma, Stealth, capace di togliere - e di reinserire nella fase di ricostruzione - tutte le informazioni diverse dal blocco di dati cifrati. Il file che esce da Stealth appare come una sequenza di bit del tutto casuale, che è molto difficile distinguere da rumore ad alta entropia. Naturalmente, chiunque può provare ad applicare il procedimento inverso (prima Stealth per ricostruire l'intestazione, quindi il PGP), ma solo disponendo della chiave giusta si potrà alla fine accedere al messaggio in chiaro. In caso contrario non si potrà neppure capire se il fallimento sia dovuto al fatto di non disporre della chiave giusta oppure, verosimilmente, al fatto che l'immagine non contiene alcun messaggio nascosto.

Un metodo alternativo all'uso congiunto di PGP e Stealth è dato dall'uso di programmi espressamente progettati per trasformare un file in rumore apparente (per esempio, Wnstorm, *White Noise Storm*). Riassumendo, un sistema steganografico completo deve comprendere due fasi fondamentali: trasformazione del messaggio in chiaro in rumore apparente. Questa fase prevede l'uso di un sistema di crittografia convenzionale e quindi di un qualche tipo di chiave; iniezione nel (o generazione del) messaggio contenitore.

Un metodo alternativo: le permutazioni pseudocasuali

Per concludere, accenniamo a una tecnica iniettiva in cui le due fasi fondamentali sono intimamente intrecciate e vengono eseguite contemporanea-

mente. La tecnica è basata sul concetto di *permutazione pseudocasuale*. Come è noto, una permutazione è una corrispondenza biunivoca di un insieme in se stesso. Per esempio, se consideriamo l'insieme dei numeri interi compresi tra 0 e 4, la funzione P definita nel modo seguente:

```
P(0) = 2, P(1) = 1, P(2) = 4, P(3) = 0, P(4) = 3
```

è una permutazione (per gli amanti delle lettere piuttosto che della matematica, il concetto di permutazione si avvicina a quello di anagramma; l'idea è la stessa: si tratta di "rimescolare" le componenti di un oggetto).

Dato un insieme di M elementi, esistono M! possibili permutazioni dell'insieme: al crescere del numero degli elementi, il numero di tutte le possibili permutazioni cresce molto velocemente. Un generatore di permutazioni pseudocasuali è una funzione che prende in ingresso un numero M - la dimensione dello spazio da permutare - e una chiave segreta, e restituisce in uscita una permutazione dell'insieme {0, 1, 2, ..., M-1} apparentemente casuale, ma dipendente dalla chiave. La permutazione generata deve essere computazionalmente sicura, il che significa che non è possibile indovinare la permutazione senza conoscere la chiave.

La tecnica funziona nel modo seguente: supponiamo che i bit modificabili (per esempio, i bit meno significativi di un'immagine) del messaggio contenitore siano M e che i bit che compongono il messaggio segreto siano N (dove N≤M; se fosse N>M, il contenitore non sarebbe abbastanza capiente). Scelta una chiave e indicando con P la permutazione di {0, 1, 2, 3, ..., M-1} generata da M e da tale chiave, allora l'i-esimo bit del messaggio segreto viene sostituito al P(i)-esimo bit modificabile, e non all'i-esimo, come opererebbe una semplice tecnica sequenziale. In questo modo il messaggio segreto viene "sparpagliato" in modo apparentemente casuale all'interno del contenitore e l'unico modo per recuperarlo è quello di conoscere la chiave utilizzata per generare la permutazione.

Ma c'è di più: i bit potenzialmente utilizzabili sono M, ma di questi ne vengono effettivamente utilizzati soltanto N, i rimanenti M-N restano sicuramente inalterati; quindi, se N è strettamente minore di M, non è possibile neppure conoscere quali sono i bit effettivamente utilizzati tra i possibili candidati. L'idea cruciale è che la generazione della permutazione selezioni implicitamente il sottoinsieme dei bit utilizzabili che verranno effettivamente utilizzati (l'insieme individuato dagli indici {P(0), P(1), P(2), ..., P(N-1)}). Aumentando M, cioè la capienza del contenitore, tale sottoinsieme si apre a ventaglio riadattandosi in modo automatico nello spazio più ampio. Se N è molto minore di M, il messaggio segreto viene letteralmente "esploso" all'interno del contenitore. Come avevamo già osservato, all'utilizzo di un conte-

nitore più ampio del necessario corrisponde un aumento del grado di sicurezza del sistema, purché i bit utilizzati non siano localizzati, ma distribuiti su tutto il contenitore. Questo risultato si ottiene in modo automatico con la tecnica delle permutazioni pseudocasuali.

Dalle parole ai fatti: guida ad alcuni programmi reperibili in rete

Una buona raccolta di informazioni e di programmi relativi alla steganografia è stata creata, sotto forma di pagina Web, da Eric Milbrandt: il suo "Steganography Info and Archive" contiene un gran numero di collegamenti a documentazioni e programmi sparsi in tutto il mondo. Da tale pagina è anche possibile iscriversi alla mailing list utilizzata da Milbrandt per dare informazioni sulla disponibilità di nuovo software (o nuova documentazione) in ambito steganografico. La presente guida ha ovviamente un carattere puramente introduttivo, per avere ulteriori dettagli si rimanda alle documentazioni allegate ai vari programmi.

S-Tools (autore: Andy Brown)

S-Tools è un pacchetto shareware composto da alcuni programmi steganografici per Windows: la versione 3 comprende tre utility per Windows 3.*, la versione 4 ne comprende due per Windows 95/NT. I programmi sono: ST-BMP, che consente di iniettare un file dentro immagini BMP a 24 bit o immagini GIF a 256 colori; ST-WAV che usa come contenitori file sonori in formato WAV; ST-FDD che nasconde le informazioni nello spazio non usato dei floppy disk (quest'ultimo è presente solo nella versione 3). La tecnica steganografica utilizzata è quella di base della steganografia sostitutiva (sostituzione del bit meno significativo), integrata con alcuni algoritmi di crittografia selezionabili dall'utente e dalla capacità di distribuire uniformemente le informazioni da nascondere nel contenitore, nel caso in cui questo sia più grande del necessario.

ST-FDD individua i settori non utilizzati di un floppy disk analizzando la FAT del dischetto, senza ovviamente modificarla; pertanto, dopo aver nascosto delle informazioni con questo sistema, occorre fare molta attenzione

142	16	10	57	19	30	182	85	79	209	142	138	224	196	197
143	16	10	56	18	31	183	85	79	208	143	138	225	196	197
142	17	10	57	18	31	190	100	96	209	143	138	224	197	197
143	17	10	56	19	31	191	100	96	208	142	139	225	197	197
160	18	10	57	19	31	190	101	96	209	142	139	78	110	198
161	18	10	168	40	32	191	101	96	208	143	139	79	110	198
160	19	10	169	40	32	190	100	97	209	143	139	78	111	198
161	19	10	168	41	32	191	100	97	180	144	148	79	111	198
142	16	11	169	41	32	190	101	97	181	144	148	78	110	199
143	16	11	168	40	33	191	101	97	180	145	148	79	110	199
142	17	11	169	40	33	180	104	100	181	145	148	78	111	199
143	17	11	168	41	33	181	104	100	180	144	149	79	111	199
160	18	11	169	41	33	180	105	100	181	144	149	24	78	204
161	18	11	144	50	52	181	105	100	180	145	149	25	78	204
160	19	11	145	50	52	180	104	101	181	145	149	24	79	204
161	19	11	144	51	52	181	104	101	214	162	158	25	79	204
148	20	14	145	51	52	180	105	101	215	162	158	24	78	205
149	20	14	144	50	53	181	105	101	214	163	158	25	78	205
148	21	14	145	50	53	98	70	104	215	163	158	24	79	205
149	21	14	144	51	53	99	70	104	214	162	159	25	79	205
162	22	14	145	51	53	98	71	104	215	162	159	228	212	214
163	22	14	104	50	56	99	71	104	214	163	159	229	212	214
162	23	14	105	50	56	98	70	105	215	163	159	228	213	214
163	23	14	104	51	56	99	70	105	218	172	170	229	213	214
148	20	15	105	51	56	98	71	105	219	172	170	228	212	215
149	20	15	172	62	56	99	71	105	218	173	170	229	212	215
148	21	15	173	62	56	198	120	116	219	173	170	228	213	215
149	21	15	172	63	56	199	120	116	218	172	171	229	213	215
162	22	15	173	63	56	198	121	116	219	172	171	240	230	228
163	22	15	104	50	57	199	121	116	218	173	171	241	230	228
162	23	15	105	50	57	198	120	117	219	173	171	240	231	228
163	23	15	104	51	57	199	120	117	224	184	182	241	231	228
162	26	16	105	51	57	198	121	117	225	184	182	240	230	229
163	26	16	172	62	57	199	121	117	224	185	182	241	230	229
162	27	16	173	62	57	54	60	118	225	185	182	240	231	229
163	27	16	172	63	57	55	60	118	224	184	183	241	231	229
162	26	17	173	63	57	54	61	118	225	184	183	248	246	246
163	26	17	180	72	66	55	61	118	224	185	183	249	246	246
162	27	17	181	72	66	54	60	119	225	185	183	248	247	246
163	27	17	180	73	66	55	60	119	184	176	186	249	247	246
112	20	20	181	73	66	54	61	119	185	176	186	248	246	247
113	20	20	180	72	67	55	61	119	184	177	186	249	246	247
112	21	20	181	72	67	124	104	132	185	177	186	248	247	247
113	21	20	180	73	67	125	104	132	184	176	187	249	247	247
112	20	21	181	73	67	124	105	132	185	176	187	250	250	250
113	20	21	182	84	78	125	105	132	184	177	187	251	250	250
112	21	21	183	84	78	124	104	133	185	177	187	250	251	250
113	21	21	182	85	78	125	104	133	224	196	196	251	251	250
56	18	30	183	85	78	124	105	133	225	196	196	250	250	251
57	18	30	182	84	79	125	105	133	224	197	196	251	250	251
56	19	30	183	84	79	208	142	138	225	197	196	250	251	251
												251	251	251

Tabella 1

a non accedere nuovamente al floppy con i normali comandi DOS di copia file, poiché il DOS potrebbe allocare proprio i settori su cui è stata nascosta l'informazione, distruggendola senza alcuna possibilità di recupero.

I programmi hanno un'interfaccia grafica che ne rende particolarmente facile e intuitivo l'utilizzo; inoltre è disponibile un file di aiuto richiamabile selezionando l'opportuna opzione dall'Help Menu.

Si tratta senza dubbio di un ottimo strumento, anche se occorre fare molta attenzione all'utilizzo di contenitori come le immagini a 256 colori. Come si è visto, i metodi più diffusi per l'utilizzo di formati grafici con palette come contenitori, si basano su algoritmi per la riduzione del numero dei colori della palette, allo scopo di aggiungere colori simili a quelli rimasti, riservandosi in questo modo delle possibilità di scelta per ogni pixel dell'immagine. Dalla documentazione allegata a S-Tools possiamo constatare che anche ST-BMP funziona proprio in questo modo. Anche se non disponiamo dei sorgenti di questi programmi, possiamo fare un semplice esperimento per capire meglio come tutto questo possa influire sulla sicurezza del sistema.

L'archivio s-tools3.zip comprende, oltre ai programmi veri e propri, due file audio .wav e una immagine in formato GIF; dentro l'immagine (così come dentro uno dei file audio) è stato iniettato un messaggio, che è possibile leggere utilizzando i programmi inclusi nel pacchetto. Abbiamo innanzitutto verificato che l'immagine utilizzi effettivamente tutti i 256 colori presenti nella palette, quindi ci siamo fatti riordinare tali colori rispetto all'ordinamento sulle componenti RGB e ciò che abbiamo ottenuto è riportato nella Tabella 1.

Considerando i primi 16 colori della tabella, suddivisi in 4 gruppi di 4 ciascuno, possiamo facilmente osservare che ogni colore differisce in misura minima dagli altri dello stesso gruppo (Tabella 2). Inoltre, accorpando il primo gruppo con il terzo e il secondo con il quarto, otteniamo 2 gruppi di 8 colori, ciascuno dei quali è molto simile agli altri colori dello stesso gruppo. Questo procedimento può essere ripetuto per tutti gli altri colori della palette, ottenendo così 32 gruppi di 8 colori ciascuno. È facile capire, quindi, che S-Tools in una prima fase riduce effettivamente a 32 il numero dei colori utilizzati dall'immagine, dopodiché può sfruttare la ridondanza che si è ve-

Tabella 2			
142 16 10	160 18 10	142 16 11	160 18 11
143 16 10	161 18 10	143 16 11	161 18 11
142 17 10	160 19 10	142 17 11	160 19 11
143 17 10	161 19 10	143 17 11	161 19 11

nuta a creare nella palette per associare a ogni pixel tre bit di informazione, proprio come se si trattasse di un'immagine true color a 24 bit. Se si considera lo spazio di tutti i colori possibili come un cubo di lato 256 (sulle 3 dimensioni abbiamo i livelli di rosso, verde e blu, rispettivamente), si può vedere che i gruppi di cui abbiamo parlato non sono altro, in questo modello, che cubetti di lato 2 contenuti nel cubo completo (un cubo di lato l comprende un volume di l^3 colori.)

Senza entrare in ulteriori dettagli matematici, ci limitiamo a osservare che è possibile scrivere un semplice programma che, presa una immagine in ingresso, effettui un'analisi del tutto analoga a quella appena descritta: se è possibile ripartire l'insieme dei colori in gruppi di colori simili, allora è molto probabile che l'immagine contenga informazione steganografata. Sebbene l'algoritmo usato da ST-BMP per ridurre il numero dei colori sia effettivamente molto buono (nel senso che l'immagine ottenuta risulta poco distinguibile ad occhio nudo da quella originale), la conclusione più prudente è che non conviene usare per scopi steganografici questo programma (e naturalmente tutti i programmi che usano tecniche analoghe) con immagini GIF come contenitori. Occorre fare attenzione a non farsi trarre in inganno dal fatto che una preventiva crittazione del messaggio segreto possa comunque resistere a tutti gli attacchi noti: un sistema steganografico è da considerarsi debole se è possibile rilevare la presenza di informazione con un alto grado di attendibilità, indipendentemente dal fatto che sia possibile o meno accedere al contenuto vero e proprio dell'informazione. Lo stesso problema non si presenta utilizzando ST-BMP con immagini BMP a 24 bit, e neppure con gli altri programmi inclusi nel pacchetto.

Psteg (autore: Roberto Fabbri)

Psteg è una libreria che implementa la tecnica delle permutazioni pseudocasuali discussa nel presente capitolo e quindi comprende soltanto il primo stadio di un sistema steganografico completo. La chiave usata dal generatore di permutazioni può essere una normale passphrase inserita dall'utente a tempo di esecuzione, oppure una chiave pubblica selezionata dai keyrings del PGP (se installato).

La libreria consente la creazione di programmi completi a partire da algoritmi di iniezione già esistenti, fornendo loro il supporto di funzioni che provvedono a comprimere preventivamente il file e a permutare i bit del file compresso, tenendo conto delle dimensioni del contenitore. Nel pacchetto sono inclusi due esempi già compilati per DOS, che utilizzano la libreria:

CJPEG/DJPEG (si tratta dei classici programmi dell'Independent JPEG Group), ai quali è stata aggiunta una opzione per iniettare/estrarre un file in una immagine JPEG, e WAVPSTEG, che permette di iniettare/estrarre un file in un suono wave. Inoltre vengono date tutte le istruzioni necessarie per installare i sorgenti in C.

Vediamo un esempio di utilizzo dei programmi CJPEG/DJPEG. Supponiamo di voler iniettare nell'immagine superman.gif (ma potrebbe essere anche in formato TARGA o in uno degli altri formati supportati dal programma) il file luthor.txt in modo che solo Joe Lametta possa leggerlo; il comando da dare è:

```
cjpeg -psteg lametta superman.gif superman.jpg < luthor.txt
```

In questo modo viene creato il file superman.jpg, che può essere inviato a Joe Lametta, il quale a sua volta può recuperare il file originale mediante il comando:

```
djpeg -psteg lametta superman.jpg superman.pbm > lex.txt
```

Joe Lametta, dopo aver battuto correttamente la propria passphrase, potrà trovare il messaggio segreto nel file lex.txt.

Texto (autore: Kevin Maher)

Questo programma trasforma file di testo uuencodati (o ASCII-armoured del PGP) in una sequenza di frasi inglesi, implementando una steganografia di tipo generativo, dal momento che il testo generato non esiste in precedenza. Il programma trae ispirazione dai Mad Libs (improvvisazioni demenziali), un gioco che andava di moda un po' di anni fa negli Stati Uniti. Ogni Mad Lib era un raccontino breve del tutto banale, salvo il fatto di essere incompleto. In ogni racconto mancavano certe parole, lasciando una sequenza di spazi indicati solo con la parte del discorso (nome, verbo, aggettivo, avverbio, ecc.) sintatticamente richiesta per completare la frase. Chi conduceva il gioco chiedeva semplicemente agli altri di dire il primo sostantivo, verbo ecc. che venisse alla mente, inserendoli quindi nelle lacune del racconto. I giocatori non sapevano quale poteva essere il racconto, sapevano soltanto che dovevano fornire al conduttore certe parti del discorso. È facile immaginare che i racconti ottenuti in questo modo, sebbene fossero percepiti come sintatticamente corretti, suonavano del tutto ridicoli a causa delle incoerenze semantiche che si venivano a creare.

Il programma Texto funziona in modo del tutto analogo ai Mad Libs. Innanzitutto ecco un semplice esempio di utilizzo. Con un text editor si crei un file, superman.txt, contenente il messaggio segreto "Morte a Superman!", dopodiché lo uuencodiamo con le apposite utility ottenendo il file superman.uue. A questo punto, dando il comando:

```
texto superman.uue
```

quello che otteniamo in uscita è il seguente messaggio:

```
The watch smartly places to the pathetic road. I mangle idle
shoes near the plastic yellow stadium.Sometimes, frogs count
behind solid windows, unless they're blue.Never lean fully
while you're rolling through a wooden sticker.We familiarly
train around green powerful rooms.While candles wanly grasp,
the units.
```

Il funzionamento è abbastanza semplice: l'autore del programma ha preventivamente preparato un file WORDS che contiene un piccolo dizionario di parole inglesi, suddivise in cinque categorie sintattiche, ciascuna contenente 64 parole. Le categorie sono: oggetti, luoghi, verbi, avverbi e aggettivi. Il file STRUCTS, anch'esso preventivamente preparato dall'autore, contiene una lista di frasi preconfezionate ma incomplete (template), contenenti "buchi" da riempire con opportune parole scelte dal precedente dizionario. Un esempio di template è il seguente:

```
Sometimes, _THINGs _VERB behind _ADJECTIVE _PLACEs, unless they're
_ADJECTIVE.
```

I "buchi" sono le parole che iniziano con il carattere "_" seguito da una delle cinque categorie sintattiche. Una volta scelto un template in modo casuale, il programma può ricavarne una vera e propria frase - più o meno sensata, come si è visto, ma sintatticamente corretta - riempiendo i buchi con parole scelte dal dizionario e appartenenti alle categorie specificate. Dato che per ogni categoria esistono 64 alternative, ogni scelta può codificare sei bit del file segreto. Per esempio, il template precedente contiene cinque "buchi" e quindi può codificare fino a 30 bit.

La fase di decodifica è implementata in maniera estremamente semplice: il testo viene scandito alla ricerca della prima parola contenuta nel dizionario; quando ne viene trovata una è possibile ricostruire i primi sei bit di informazione in base alla posizione occupata dalla parola nel dizionario: se la parola occupa la prima posizione, i bit estratti sono 000000, se è la seconda 000001, ecc., fino ad arrivare alla sessantaquattresima che codifica la sequenza 111111. Si continua con la ricerca della seconda parola, e così via, fino alla fine del testo (si noti che la semplicità di questa soluzione è resa possibile dal fatto

che il dizionario e i template non contengono parole in comune; in caso contrario il programma dovrebbe essere in grado di effettuare una sorta di "pattern matching" tra i template e le frasi del testo.)

Si osservi che questo metodo funziona solo se il mittente e il destinatario usano lo stesso dizionario - stesse parole nello stesso ordine - mentre la lista dei template può essere modificata a piacere (purché si rispettino i vincoli discussi nella precedente nota). Questo significa che ognuno può divertirsi a creare diverse versioni dei file WORDS e STRUCTS - che sono semplici file di testo - con parole e template diversi e in diverse lingue, senza la necessità di modificare il programma eseguibile. Ciò che conta è che il mittente e il destinatario si accordino preventivamente sullo stesso file WORDS da usare.

Impiegando nella pratica questo sistema, ci si rende conto abbastanza presto di quali siano i suoi limiti: anche dotando il file STRUCTS di un numero molto elevato di template non si può fare a meno di notare il modo innaturale in cui si ripetono gli stessi schemi e le stesse parole, per non parlare della quasi totale assenza di senso compiuto che caratterizza i testi generati. È impensabile, quindi, che questo metodo non possa insospettire un eventuale controllore umano che abbia la possibilità di intercettare il messaggio contenitore.

Tuttavia un programmino come questo, pur nella sua semplicità, è in grado di mettere in crisi il tentativo di progettare un metodo automatico di attacco nei suoi confronti. Infatti le abilità necessarie per capire che un testo può essere generato da un programma come questo non sono ottenibili mediante semplici manipolazioni sintattiche (la semplice rilevazione di parole che occorrono varie volte non è sicuramente un criterio sufficiente), ma richiedono processi di percezione di alto livello del tutto comuni nelle persone e che tuttavia, al presente stato della ricerca in intelligenza artificiale, non sono ancora ben compresi né tantomeno disponibili sotto forma di modelli computazionali.

Resta da osservare che, per un corretto uso di questo programma, è necessario crittare preventivamente il messaggio segreto e trasformare il file crittato in rumore apparente, eliminando eventuali segni identificativi o applicando uno degli altri metodi che abbiamo discusso.

Possiamo muovere una critica a Texto: perché limitarsi ad accettare in ingresso solo file uuencodati (o ASCII-armoured del PGP)? Il fatto che gli insiemi di caratteri usati da questi tipi di file consistano esattamente di 64 elementi - un numero che nella pratica appare ragionevole adottare per le alternative di ogni categoria sintattica - non è una ragione valida, dal mo-

mento che qualsiasi file può essere visto come una sequenza lineare di bit. Di Texto sono anche disponibili i sorgenti in C, facilmente compilabili su qualsiasi piattaforma.

Stego (autore: Andrea Mazzoleni)

Stego è un altro esempio di programma che implementa una steganografia di tipo generativo, ma il suo funzionamento interno è radicalmente diverso da quello di Texto. Esso si basa sul concetto di *funzioni mimiche*, in particolare sull'idea di imitare lo stile di scrittura di un testo preso come modello di riferimento. Per fare questo, in una prima fase analizza il testo di riferimento costruendo un *dizionario* che contiene informazioni statistiche sulle sequenze di lettere che compaiono nel testo; in un secondo tempo, utilizza sia il file segreto che il dizionario per generare un testo avente le stesse caratteristiche statistiche del testo di riferimento. Il testo generato in questo modo, tuttavia, in genere non avrà un senso compiuto nemmeno al livello delle parole (si può parlare di pseudo-parole, cioè parole inesistenti che "suonano" come parole reali, o meglio come le parole che compaiono nel testo preso come modello). Tuttavia è anche possibile raccogliere informazioni sulle sequenze di parole invece che di lettere; in questo caso il testo sarà lessicalmente corretto, ma continuerà a non avere senso compiuto.

Il programma ha un'opzione molto interessante che consente di ottenere un testo che può essere decodificato senza disporre del testo di riferimento usato per creare il dizionario. Questa soluzione evita alle due parti che vogliono comunicare di doversi preventivamente accordare sul testo di riferimento (con tutti i problemi pratici che un tale accordo preventivo potrebbe comportare).

Vediamo più in dettaglio il funzionamento della fase di codifica attraverso le funzioni mimiche. Si prende un testo di riferimento e si misurano le occorrenze delle sequenze di testo lunghe N, con N scelto a piacere. Ad esempio con N=4 e il testo "questa è una questura", abbiamo:

```
sequenza occorrenza
"ques"   2
"uest"   2
"esta"   1
"sta "   1
"ta e"   1
"a e'"   1
...
```

```
"estu"   1
"stur"   1
"tura"   1
```

A questo punto si procede alla generazione del testo. Si considera il file da steganografare come uno stream di bit 0 e 1 distribuiti uniformemente da cui si estrae un bit alla volta. Supponendo il processo già a regime si guardano le ultime N-1 lettere generate del testo steganografato e si sceglie la lettera successiva in funzione dei bit letti dallo stream. Ad esempio con le N-1 lettere uguali a "est" ci sono due sequenze possibili:

```
"estu"   1 50%
"esta"   1 50%
=============
         2 100%
```

quindi si estrae un bit; se è 0 si genera la lettera "u", se è 1 si genera la lettera "a". Supponendo che sia il bit 1 le N-1 lettere diventano "sta" ed il processo si ripete. Naturalmente se ci sono più di due sequenze possibili il numero di bit codificati aumenta. Ciò si ottiene con una codifica simile a quella di Huffman. Ad esempio, le sequenze:

```
seq.     perc.
"estu"   5%
"esta"   10%
"estr"   10%
"esti"   40%
"esto"   20%
"este"   15%
```

vengono partizionate in due gruppi in modo da avere circa il 50% da ciascuna parte:

```
seq.     perc.    bit
"esti"   40%      0
"estr"   10%      0
"esto"   20%      1
"este"   15%      1
"esta"   10%      1
"estu"   5%       1
```

e il processo viene ripetuto fino ad avere un codice per ogni sequenza, con l'accortezza di generare i codici solo se la partizione è circa 50%-50%.

Nel nostro esempio, ad "esti" (40%) ed "estr" (10%) la generazione di bit viene fermata e a entrambi i simboli viene assegnato il codice 0. Questo per evitare che nel testo steganografato ci siano troppe sequenze "estr" rispetto alle sequenze "esti", cioè un rapporto 1:1 invece che 1:4, rendendo il testo

non più molto simile a quello di riferimento. L'effettivo simbolo viene invece scelto casualmente, ma dato che hanno tutti lo stesso codice, la decodifica rimane comunque possibile. Questo modo di procedere rende la codifica non deterministica, cioè codificando più volte lo stesso file si otterranno codifiche differenti ma equivalenti.

La decodifica senza dizionario deriva dall'associazione di una funzione di hash e della modalità di codifica attraverso le funzioni mimiche. Invece che codificare il file da steganografare viene codificato uno stream di bit pilotato e in contemporanea viene analizzato con la funzione di hash il risultato della codifica. Il processo viene ripetuto con tecnica backtracing finché la funzione di hash non generi come output il file da steganografare. Nella fase di decodifica basterà quindi riapplicare la funzione di hash per riottenere il file originale.

Per quanto riguarda il problema della sicurezza, le considerazioni relative a Texto sono valide anche per questo caso: prima si utilizza un sistema di crittografia convenzionale avendo cura di farne sparire le tracce; quello che si ottiene, poi, lo si manda in ingresso a Stego. Per le stesse ragioni già discusse a proposito di Texto, è molto difficile progettare un metodo di attacco automatico a questo sistema.

Per finire vediamo qualche esempio di utilizzo. Per prima cosa creiamo i file di tipo "dizionario" contenenti le informazioni statistiche che vogliamo conservare nei testi generati. Se il file divina.txt contiene il testo della Divina Commedia, i seguenti comandi creano i file divina.l5 e divina.w3

```
stego -il divina.txt 5 -w divina.l5
stego -iw divina.txt 3 -w divina.w3
```

Il primo prende informazioni riguardo a tutte le sequenze di 5 lettere contenute nel testo della Divina Commedia, il secondo considera invece sequenze di 3 parole. Utilizziamo lo stesso messaggio segreto "Morte a Superman!" dell'esempio precedente, contenuto nel file superman.txt, e diamo il comando:

```
stego -S 16 -d divina.l5 -f superman.tex superman.txt
```

Il testo che otteniamo in uscita (file superman.tex) è il seguente:

```
Surge ad Arli, e' tolto. Vinci e staglie o difesa, pur ch'i' ne poi di
Mirra sportero' guardi lieve cosa m'hanno ne' grembo e guarda col fiume
fia che 'l ciel terza si` la ferme al romore dal fornisce e bella luce
e' tonda tal colle sue vena e 'l maestro questa punge, qua nuovo li
guasti
```

Utilizzando il secondo dizionario otteniamo:

```
stego -S 16 -d divina.w3 -f superman.tex superman.txt
```

```
Temi e vederai l'altro disse, quella c'e' parte una persona li caldi e
onde l'atterra torto e io fui del mal di Taumante, che le labbra mie
vene e rompe i passi ad una, o ver lui che tiene ancor le due le sta
come due si movono a Dio: 'Piu' mi trassi. Ed ei mi bagna. Io dicea con
la man
```

si osservi che, a causa del non-determinismo nel processo di codifica, ripetendo più volte lo stesso comando, tipicamente si ottengono risultati diversi. Applicando ad entrambi i testi il comando:

```
stego -X 16 -f superman.out superman.tex
```

si ottiene nuovamente il messaggio originale (leggibile nel file superman.out). In questi due esempi abbiamo usato le opzioni che utilizzano la funzione di hash (-S e -X maiuscole); questo ha reso possibile la decodifica senza specificare un dizionario. La distribuzione di Stego comprende anche i sorgenti in C++, compilabili su diversi sistemi.

Conclusioni

La disponibilità delle tecniche steganografiche discusse in questo capitolo ha una immediata conseguenza: *dovunque esista un canale di comunicazione è impossibile controllare le informazioni che vi transitano.* La sola disponibilità della crittografia convenzionale non implica l'impossibilità di esercitare una forma di controllo sul canale: si può pensare a casi reali in cui l'uso della crittografia non sia ben visto o sia addirittura proibito.

Da tutto quello che abbiamo visto, e considerando la semplicità con cui qualsiasi persona avente modeste capacità di programmazione è in grado di mettere a punto sistemi steganografici estremamente efficaci, si può concludere che la sola idea di "proibire l'uso della crittografia" è assai ingenua, poiché ignora completamente la reale natura delle informazioni. Allo stesso modo attuare una simile proibizione, per legge o regolamento interno, sarebbe del tutto inutile, data l'impossibilità pratica di controllare che il divieto venga effettivamente rispettato.

C'è un solo modo, palesemente assurdo e praticamente impossibile, per proibire la privacy nelle comunicazioni: abolire del tutto e definitivamente *qualsiasi* comunicazione.

Steganografia è un altro parolone, amico. Ma credi a me, a chi ha avuto questa bella idea bisognerebbe fargli un monumento.

Perché se ci pensi tutta la faccenda della crittografia funziona benissimo, ma un difettuccio ce l'ha pure quella, come ti dicevo. Questa roba crittata la puoi riconoscere a colpo d'occhio e volendo la blocchi con facilità... Beh, forse hai ragione amico, proprio così facile non è. Per adesso. Ma se qualche pezzo grosso, di quelli grossi sul serio, dice che una certa cosa è fuorilegge, vedi tu se non può diventare un bel casino. Non per noi, ovviamente, che l'unica legge che riconosciamo è quella che ti dice di vivere una volta sola. Ma di gente come noi, ti ripeto, ne è rimasta poca, e se ci troviamo da soli a fare certe cose rischiamo di dare un po' troppo nell'occhio. Ci ha pensato pure Superman che in questo modo poteva metterci in un bel casino. Allora ha preso su e si è fiondato direttamente alla Casa Bianca. Anzi, DENTRO la Casa Bianca. A mezzanotte, lasciandosi dietro un altro bel po' di buchi con la sua sagoma nei muri. Era sempre più incazzato il nostro Omino d'Acciaio, amico, e del conto della spesa per richiudere tutte quelle nuove aperture per la ventilazione non gliene fotteva mica più di tanto. E nemmeno del cerimoniale per le visite al Presidente degli Stati Uniti, direi. Niente guardia d'onore schierata, televisione, salamelecchi e bandiere americane al vento, stavolta. Un sibilo, una serie di botti e Superman era nella stanza da letto dell'Appartamento Presidenziale. Il buon vecchio Bill si è preso una strizza della madonna, lì per lì ha pensato a una bomba, o all'incursione di un commando di terroristi arabi. Ma quando il nuvolone di polvere si è diradato e ha visto che era Superman in piedi su un mucchio di calcinacci, ha aperto la bocca per dirgliene quattro. Che cazzo fai a fare il Presidente degli Stati Uniti d'America se il primo SuperCoglione che passa può venirti a rompere le palle a casa tua senza neanche un appuntamento, peggio che all'ultimo degli idraulici? Ma ha fissato quegli occhi rossi come i fanalini di coda di una Cadillac e ha pensato bene di lasciar perdere. Perché in quell'ammasso di muscoli che gli stava ritto davanti in silenzio, c'era qualcosa di minaccioso, anche se l'idea era incredibile. Strano però, non se n'era mai accorto prima di quanto era GROSSO Superman. Preoccupante. Per cui ha richiuso la bocca inghiotten-

do un bel po' di saliva, ha fatto segno di star buoni agli scagnozzi dei Servizi che stavano per dar fuori di matto e ha aspettato di vedere come si mettevano le cose con quella specie di locomotiva impazzita. "Dolente di disturbarla nell'adempimento dei suoi doveri, signor Presidente" ha cominciato Superman "ma si tratta di un'emergenza assolutamente prioritaria per la Sicurezza Nazionale. È indispensabile che Lei metta immediatamente fuorilegge l'uso di qualunque forma di crittografia su tutto il pianeta". "Caro SuperAmico" gli ha risposto allora sollevato il vecchio Bill "se non si tratta che di questo, ci stiamo già lavorando da un pezzo. Mi dia un po' di tempo, domani solleverò la questione al congresso e vedrà che nel giro di qualche mese..." "Temo di non essermi spiegato signor Presidente" l'ha interrotto Superman "ho detto IMMEDIATAMENTE". "Mio caro amico" ha ripreso il presidente, un po' preoccupato a quel punto "forse lei non si rende conto dei problemi: il Congresso, le complicazioni internazionali, le giurisdizioni degli altri paesi…" "Penserò io a far rispettare la volontà degli Stati Uniti d'America nel mondo, signor Presidente" ha risposto Superman. "Lei mi dia solo l'ordine scritto." E ha appoggiato la manona sulla scrivania del vecchio Bill, facendo un'impronta profonda un buon cinque centimetri nel mogano. Ora, Billy non è precisamente un pivellino nemmeno lui, a modo suo. Se sei abituato ad abbrustolire città intere con una telefonata, mica puoi essere tanto tenero, amico. Ma fissando quell'impronta ha scribacchiato qualche riga su un foglio, l'ha firmato e l'ha dato senza dire una parola a Superman. Che è decollato immediatamente lasciandosi dietro un altro bel po' di lavoro per i muratori, piantandolo lì come un allocco. E senza nemmeno fare il saluto militare, stavolta. Dopo un'oretta di trattative diplomatiche condotte alla sua maniera, in rete non potevi più trovare un messaggio crittato nemmeno a cercarlo col lanternino. Tranne i nostri beninteso. Era un vero spasso. Grazie alla steganografia iniettavamo i nostri messaggi nelle foto delle pupe, nelle canzoni, in quello che ci pareva. Nelle pagine web, ad esempio. Uno pensa di trovarsi di fronte alla pagina web dell'Associazione Amici dell'Uncinetto e invece chissà cosa c'è nascosto in mezzo. Superman lo sapeva che continuavamo a tenerci in contatto: ormai lo scadere dell'ultimatum per la Bomba era vicino. E a un certo punto ha pure mangiato la foglia. Ma non poteva mica mettersi a controllare tutte le immagini, i suoni e palle varie che passano per la rete, mi capisci? E anche se l'avesse fatto, senza chiave non ci cavava comunque un ragno dal buco: erano tutti uguali. Era solo questione di tempo e di un po' di fantasia, e anche tutti i remailer e nymserver si sarebbero presto adeguati alla nuova situazione. Però una cosa bisogna dirla: il nostro Omino d'Acciaio è uno che non molla facilmente. Nossignori. Ma nemmeno il qui presente, amico. Ragion per cui, invece di sedermi sugli allori, mi sono affrettato a imparare qualche altra coserella. Poteva tornar comoda.

Telefonia Digitale Crittata

Di Zeus Kissakie'

Crittare le telefonate: perché?

Le conversazioni che si svolgono attraverso un telefono sono molto meno private di quanto generalmente si pensi: è relativamente facile per chiunque poter ascoltare quel che passa per una linea telefonica e lo è ancora di più se parliamo attraverso un telefono cellulare. Anche i media ufficiali negli ultimi tempi hanno ampiamente parlato di intercettazioni telefoniche, di linee sotto controllo, di cellulari captati con gli scanner, di boss mafiosi più o meno famosi traditi da una telefonata: si badi che le intercettazioni non avvengono soltanto in seguito a interventi di un magistrato in odore di carriera, ma c'è un discreto numero di persone che vi dedica parte del proprio tempo: non solo agenzie investigative, ma anche curiosoni della domenica e phone phreaker di tutte le età. È relativamente facile intercettare le comunicazioni telefoniche: ora che le linee sono quasi tutte digitali, spesso non c'è nemmeno bisogno di collegarsi fisicamente alla linea, come una volta si faceva con i classici morsetti a coccodrillo; negli USA, ad esempio, basta un codice di accesso, magari rubato, magari trovato scartabellando tra la spazzatura delle società telefoniche, ed è possibile ascoltare telefonate altrui dall'altro capo della nazione; tutto lascia presupporre che sia possibile fare altrettanto anche in Italia. Per quel che riguarda la telefonia cellulare, la situazione è ancora peggiore: moltissimi telefonini, specie i vecchi modelli TACS ancora largamente diffusi, sono intercettabili con apparecchi radio di basso costo e dav-

vero alla portata di chiunque. Come se non bastasse, la tecnologia nel riconoscimento vocale ha fatto passi da gigante negli ultimi anni: i servizi segreti sono in grado di monitorare su larga scala le conversazioni telefoniche, alla ricerca di "parole" interessanti (un po' come avviene per il monitoraggio dei messaggi su Internet): non è provato che lo facciano, o che lo facciano sempre, ma hanno a disposizione la tecnologia necessaria.

Per difendersi dalle intercettazioni già negli anni 70 sono nati gli scrambler telefonici: apparecchi atti a crittare la voce umana da un capo della linea telefonica, e a riconvertirla in chiaro dall'altro capo. Il principio di funzionamento è semplice ed è simile per tutti i modelli di scrambler "ad inversione", ovvero che miscelano invertendo. Poiché la voce umana copre una banda che va da 300 Hz a 3000 Hz, è sufficiente miscelare le frequenze delle note vocali con una frequenza fissa, generata da un oscillatore e concordata prima di iniziare la conversazione. Miscelando la voce con la frequenza fissa, che in pratica è una vera e propria chiave di cifratura, otterremo per differenza un suono completamente diverso dall'originario; questo sarà assolutamente inintelligibile, a meno di non utilizzare un altro scanner che funzioni alla rovescia, e ovviamente di conoscere la chiave.

Gli scrambler di questo tipo non hanno avuto molta fortuna. In primo luogo è necessario che entrambi gli interlocutori si dotino dell'apparecchio, acquistandolo o autocostruendolo; secondariamente, la codifica non è affatto sicura: registrando la conversazione e munendosi di un apparecchio uguale è semplice trovare la chiave, anche per tentativi, essendo esigua la quantità di chiavi disponibili. Lo schema elettrico e le istruzioni per il montaggio di uno scrambler di questo tipo sono stati pubblicati in Italia dalla rivista Nuova Elettronica nel numero di giugno 1984. Il modello in questione può funzionare con soltanto sedici chiavi differenti, quindi può essere decrittato con estrema facilità. Desiderando acquistarne uno già montato e collaudato, è possibile rivolgersi a *Ramsey Electronics*, che ne vende uno per meno di 80 dollari.

Da notare che ancora oggi molte radio della polizia, quando non trasmettono in chiaro, utilizzano scrambler di questo tipo e quindi sono facilmente intercettabili. Scrambler più recenti - e più costosi - hanno un sistema di variazione continua della frequenza chiave, come quello fabbricato dalla Transcrypt (Nebraska, USA); questi modelli risultano senz'altro più sicuri di quelli a chiave fissa, pur essendo sempre vulnerabili a un attacco di forza bruta sferrato utilizzando un apparecchio simile. Un altro modello di scrambler "moderno" è quello ideato dalla Communication Security Corporation, che utilizzerebbe il Triple DES e il Diffie-Hellman (algoritmi largamente utilizzati in crittografia). Il prezzo è di "soli" 995 dollari l'uno: decisamente troppo

caro, considerando che ne occorreranno almeno due e che la sicurezza è solo sulla carta - anzi, sulla rete…

Il metodo più economico, sicuro ed efficiente di effettuare telefonate cifrate, oggi, è senz'altro munirsi di un apposito software per PC, supponendo di possedere già accessori sempre più diffusi quali modem, scheda sonora, nonché ovviamente un microfono e un paio di cuffie. Il software si occupa di digitalizzare la voce proveniente dal microfono in dati che vengono poi subito compressi, crittati e trasmessi via modem. La nostra voce diventa quindi una sequenza di bit che, dopo compressione e cifratura, sono pronti per essere inviati al nostro interlocutore. Questi programmi sono molto più efficaci dei vecchi scrambler: ci danno la possibilità di crittare le nostre telefonate con sistemi di crittografia robusta, non decifrabile da chi non è in possesso della chiave necessaria, se non disponendo per anni di un potente computer dedito soltanto alla decrittazione. In alcuni casi nemmeno molti anni di lavoro-macchina sono sufficienti, il che ci rende ragionevolmente tranquilli. E il fatto che l'esportazione di questi programmi dagli USA sia paragonata in tutto e per tutto all'esportazione di munizioni, la dice lunga sulla loro implicita potenza.

Disponendo di una connessione alla rete è inoltre possibile evitare telefonate a lunga distanza, collegandosi al più vicino Internet provider e inviando i pacchetti direttamente all'indirizzo IP dell'interlocutore, e viceversa. In pratica, se Joe Lametta, da Metropolis, vuole parlare con Lex Luthor in vacanza alle Bahamas, non ci sarà nemmeno bisogno di una telefonata intercontinentale: Joe si collegherà al proprio solito fornitore di accesso Internet a Metropolis e lo stesso farà Luthor dal luogo in cui si trova - entrambi pagheranno quindi solo la tariffa urbana; la voce di Joe Lametta verrà digitalizzata dalla scheda audio, crittata dal software e infine inviata dal modem al provider urbano. Da Metropolis alle Bahamas la voce di Joe correrà attraverso Internet e una volta giunta al provider di Luthor subirà il corrispondente processo di decodifica fino a tornare di nuovo udibile dalla cuffie di Lex. Il meccanismo funziona e grazie alla crittografia la riservatezza è assicurata, anche se in pratica le telefonate via Internet lasciano un po' a desiderare in fatto di qualità audio e velocità di risposta...

Comunque, citando Philip Zimmermann, autore di PGPfone, uno di questi programmi, "esso vi permette di sussurrare nell'orecchio di qualcuno, anche se quell'orecchio si trova a migliaia di chilometri, senza essere uditi da terze persone".

L'utilizzo della rete per telefonate a lunga distanza non è argomento da sottovalutare: anche le compagnie telefoniche hanno compreso che con la

tecnologia recente le telefonate interurbane e internazionali verranno prima o poi spazzate via da quelle effettuate tramite Internet, tariffate come urbane; ciò comporterà per loro una notevole diminuzione di guadagno e stanno cercando di correre ai ripari come meglio possono: diminuendo le tariffe, in quegli stati dove vi è un regime di concorrenza; spingendo i governi a introdurre una tassa a loro favore, dove vi è monopolio. In Italia la Telecom, che ha avuto il monopolio delle telecomunicazioni per decenni ed è solo ora sul punto di perderlo, sta ribassando le tariffe internazionali, lasciando praticamente inalterate le urbane. Il fatto che una telefonata tra due località distanti più di 30 Km sia tariffata esattamente quanto una tra località distanti 1500 Km la dice lunga, soprattutto se consideriamo che alla Telecom entrambe le telefonate costano la stessa cifra: i commutatori sono ormai per la maggior parte digitali e meno soggetti a guasti di quelli analogici, utilizzati negli anni 80, quando la differenza di tariffazione tra chiamate urbane e interurbane era, se non accettabile, per lo meno comprensibile.

Dal punto di vista legislativo, l'orientamento più recente pare voler distinguere tra telefonate vere e proprie e telefonate effettuate via Internet: nel primo caso a viaggiare è soltanto la voce degli interlocutori, nel secondo sono pacchetti di byte, che possono contenere tanto voce digitalizzata (ad esempio i file .wav) quanto dati di altro tipo, indifferentemente. Ad avvalorare questa tesi è intervenuta recentemente la Commissione Europea di Bruxelles, affermando che i servizi telefonici sulla rete Internet non possono essere definiti servizi di telefonia vocale propriamente detta. Essendo l'argomento di estrema attualità è comunque bene attendere disposizioni di legge più precise. Nel frattempo, teniamo presente che le conversazioni telefoniche "sono personali, sono private, e non sono affari di nessuno se non nostri. Potremmo pianificare una campagna politica, discutere delle tasse, o avere una relazione clandestina. O potremmo stare facendo qualcosa che non dovrebbe essere illegale, ma lo è. Di qualunque argomento trattino, non vogliamo che le nostre telefonate siano intercettate o ascoltate da qualcun altro. Non c'è nulla di male nel voler affermare il nostro diritto alla privacy" (ancora Philip Zimmermann).

Speak Freely

In italiano significa "Parla liberamente" ma anche "Parla gratuitamente": Speak Freely è distribuito al pubblico dominio e i sorgenti sono disponibili. Quest'ultima caratteristica è particolarmente apprezzata nei programmi di

crittografia, in quanto permette di verificare che non ci siano backdoor o bug grossolani. D'altronde l'autore, John Walker, non è l'ultimo studentello appena uscito dall'università: è stato fondatore della Autodesk nonché coautore di Autocad, uno dei software di CAD più diffusi – e anche più copiati. Secondo Scientific American, John Walker sarebbe anche stato riconosciuto quale l'autore del primo virus per computer, scritto nel 1975.

Il programma è disponibile per le piattaforme Windows (3.11, 95/98, NT), nonché per Unix. Le richieste hardware non sono particolarmente esose (è necessario un 486 ad almeno 50Mhz, o meglio un Pentium) anche se è noto che con un computer veloce e con tanta memoria è possibile eseguire più applicazioni contemporaneamente e lavorare con maggiore comodità. Pur essendo la telefonia crittata il principale obiettivo di questo software, è da segnalare l'esistenza di una versione senza questa caratteristica, detta "Spook Freely", nata per essere utilizzata in quelle nazioni in cui crittare le conversazioni è illegale (sebbene, paradossalmente, proprio in quelle nazioni la crittografia è probabilmente *ancora* più necessaria). Dispone di compressione audio, tale da poter essere utilizzato efficacemente con modem "lenti" quali i 14.400 bps; è ovvio che in questo caso ci deve essere un compenso della velocità da parte del processore.

Particolare attenzione è stata rivolta alla conversazione tra più persone, possibile però soltanto se si dispone di un adeguato socket di connessione a Internet: nei casi peggiori sarà sempre possibile il broadcasting, cioè uno parla e tanti ascoltano; solo disponendo di un adeguato windows socket si potrà organizzare una multiconferenza vera e propria (multicasting: tutti possono ascoltare e parlare).

Speak Freely può funzionare in due modalità: *normal* e *voice activated*. Se si utilizza la prima, consigliata ai principianti, è necessario premere il tasto destro del mouse quando si parla, ed è buona norma terminare la frase con la parola "passo" ("over" in inglese). La modalità voice activated, invece, serve a risparmiare banda di trasmissione: poiché è inutile e dispendioso in termini di banda trasmettere dati continuamente, anche quando si sta in silenzio, il voice activated permette di rilevare automaticamente se si sta dicendo qualcosa al microfono e, in quel caso, trasmetterla. Lo svantaggio è che questa opzione è più difficoltosa da configurare, è necessario eseguire prima alcune prove e determinare il giusto volume di attivazione della voce. Bisogna infatti far sì che questo non sia troppo basso, perché altrimenti la trasmissione sarebbe sempre attivata dal rumore di fondo; analogamente il livello non deve essere troppo alto, perché altrimenti anche parlando a voce alta non si attiverebbe mai la trasmissione. Da notare che la modalità voice activated non è utilizzabile con le casse, ma solo se si dispone di cuffie: i

normali speaker, ricevendo, potrebbero attivare nuovamente il microfono, generando un equivalente dell'effetto larsen e impedendo in pratica la conversazione.

È inoltre necessario prestare attenzione al modello di scheda audio che si possiede: non tutte possono trasmettere e ricevere contemporaneamente (*full duplex*) ma solo l'una o l'altra cosa per volta (*half duplex*). Le generiche schede audio a 8 e 16 bit, ad esempio, sono half duplex: in questo caso è sconsigliato l'utilizzo dell'attivazione vocale. È buona norma comunque controllare sul sito del produttore della scheda audio l'uscita di driver aggiornati, che a volte permettono buone prestazioni anche con schede sonore obsolete.

L'utilizzo della modalità full duplex è tuttavia quasi impossibile a meno di non disporre di una rete ad alta velocità (e cioè una rete locale); con una rete veloce c'è pochissimo ritardo tra quando una persona parla e quando l'altra riceve. Con una rete lenta, quale è generalmente Internet, ci può essere un ritardo sostanziale (già due o tre secondi sono fastidiosi) tra la trasmissione e la ricezione. Questo ritardo è particolarmente elevato se la connessione in rete avviene via satellite e/o se la rete è molto congestionata. In una connessione Internet full duplex tra Italia e Canada abbiamo misurato un ritardo di circa cinque secondi tra trasmissione e ricezione; il risultato è stato uno sfasamento completo nonché l'impossibilità di conversare senza utilizzare la convenzione parla/passo… cioè una conversazione esattamente equivalente a quella half duplex vera e propria!

Utilizzo pratico

Dopo aver cliccato sull'icona di Speak Freely e verificato che microfono e casse siano effettivamente collegati, il primo passo è quello di testare il software: cliccando sul menù Help possiamo raggiungere l'opzione "local loopback". Comparirà una finestra che indicherà il tentativo di connessione con noi stessi. Spostando il mouse su detta finestra (si noti che il puntatore diventa una cornetta telefonica) e tenendo premuto il tasto destro (il puntatore ora diventa un orecchio) attiviamo la trasmissione. Dopo aver detto al microfono una breve frase – e aver rilasciato il tasto del mouse al termine di questa – se tutto funziona bene dovremmo risentire questa frase nelle casse.

Possiamo ora attivare la nostra connessione a Internet e testare Speak Freely su uno degli "echo server" disponibili. Questi sono dei siti che ricevono e ritrasmettono esattamente lo stesso segnale al mittente. Dal menù Connection selezionare New e digitare il nome dell'echo server. Una lista di

echo server è disponibile nel menu di help; la più aggiornata dovrebbe essere sul sito di Speak Freely; al momento in cui scriviamo sono disponibili i seguenti:

```
echo.fourmilab.ch        Svizzera
corona.itre.ncsu.edu     U.S.A. (North Carolina)
rpcp.mit.edu             U.S.A. (Massachusetts)
```

Chiaramente è meglio utilizzare un echo server "vicino" a noi, telematicamente parlando: se il nostro Internet provider dispone di una connessione veloce con gli USA gli ultimi due siti potranno essere più "veloci" di quello svizzero. Una volta realizzata la connessione comparirà una finestra analoga a quella del local loopback, solo che questa volta i dati partiranno dal nostro PC, arriveranno all'echo server e torneranno indietro, con tutti i problemi che potranno verificarsi. Ad esempio, la cosa più facile che potrebbe accadere è che la rete sia congestionata, il che potrebbe portare a ritardi tra un pacchetto e l'altro e quindi a delle pause. Quando le pause rendono inintelligibile una conversazione, lo stesso autore confessa che non c'è nulla da fare: occorre aspettare e rimandare ad un altro momento, quando la rete sarà meno trafficata. Una soluzione parziale può essere quella di aumentare la Jitter Compensation (menù Options) per raggruppare più pacchetti in uno solo e quindi renderli più intelligibili, ma spesso nemmeno questo funziona e bisogna proprio rimandare la conversazione.

Dopo aver testato che tutto funzioni bene con un echo server potremo finalmente tentare una conversazione vera e propria. Il primo problema potrebbe essere che non sappiamo con chi parlare; infatti non bisogna dimenticare che con questo genere di software è possibile conversare esclusivamente con altri utenti Internet, non è possibile "telefonare" direttamente a casa di qualcuno[1]. Quando un utente internet desidera telefonare ad una persona in una città nella quale la ditta ha un "nodo" internet, quest'ultima si occupa di realizzare la connessione locale dalla sede del suo nodo fino alla casa del chiamato (tipicamente tramite una telefonata urbana). Per questo genere di servizio la ditta chiede all'utente una tariffa che è minore di quella internazionale, e quindi ben più conveniente; analogamente la ditta paga alla compagnia telefonica solo la telefonata urbana, guadagnando la differenza tra questa e la tariffa applicata all'utente.. Se anche abbiamo un "appuntamento" con un amico, non conoscendo il suo indirizzo IP – che è quell'indirizzo composto da quattro numeri che ci identifica in modo univoco su

[1] Non è argomento della nostra ricerca, ma segnaliamo ugualmente l'esistenza di ditte che forniscono anche questo servizio (finora solo negli Stati Uniti): essendo collegate alla rete in molte città, si sovrappongono alle compagnie telefoniche.

Internet, e che di norma varia ogni volta che ci colleghiamo alla rete – è impossibile contattarlo. Il metodo più semplice è scoprire il nostro indirizzo IP, nel caso ci fosse sconosciuto (ad esempio lanciando il programma winipcfg.exe presente in Windows 95) e inviarlo per e-mail al destinatario. Un'altra soluzione potrebbe essere l'utilizzo di un *IP poster*, cioè un programma che va a scrivere il nostro indirizzo IP dinamico su una pagina web cui abbiamo accesso in lettura e scrittura. Un'ultima soluzione, integrata in Speak Freely, è utilizzare un server LWL (Look Who's Listening, cioè Guarda chi c'è in ascolto). Questi server sono in grado di "ospitare" utenti di Speak Freely e di far conoscere il loro indirizzo IP ad altri utenti che volessero mettersi in contatto con essi. Nel menù Phonebook, selezionare Edit Listing e specificare nella prima casella il nome del server LWL da utilizzare.

Alcuni server sono anche in grado di mostrare gli utenti connessi attraverso il web. Ovviamente rimandiamo al sito ufficiale di Speak Freely per ulteriori informazioni e per la lista dei server LWL aggiornata. Ricordiamo anche che ogni server è indipendente dagli altri e, se ci si collega ad uno, solo gli altri utenti di quel server ci vedranno.

Torniamo all'opzione Edit Listing del menù Phonebook. È possibile specificare negli appositi campi il nostro indirizzo e-mail, il nostro nome, il nostro numero di telefono e la città o la nazione. Ovviamente non siamo obbligati a riempire questi spazi con dati veritieri, basta che siano sufficienti a renderci riconoscibili agli occhi delle persone con cui vogliamo venire a contatto. In particolare, prestare attenzione prima di specificare il nostro numero di telefono e l'e-mail (occhio agli spammer e ai rompiscatole di ogni tipo). Queste informazioni saranno visibili soltanto se barriamo la casella "List in directory". L'altra casella, "Exact match only", serve a renderci visibili soltanto a chi fa una richiesta (query) esatta della nostra e-mail: in pratica, se è barrata, permette solo a chi conosce la nostra e-mail di contattarci. Inoltre, se è barrata, il nostro indirizzo verrà escluso da quelli listati dal server sul web. Per fare una ricerca, dal menù Phonebook selezionare Search, e specificare il server LWL e, opzionalmente, il nome o l'e-mail dell'utente desiderato. Quest'ultimo dovrà essere specificato se vogliamo trovare un utente che ha selezionato la funzione "Exact match only" vista poco sopra.

Siamo ora in grado di aprire una nuova connessione, cioè di parlare finalmente con qualcuno. La procedura è quella descritta più sopra: dal menù Connection selezionare New e specificare l'indirizzo della persona che desideriamo contattare. Se tutto è ok dovremmo essere in grado di parlare nel microfono (tenendo premuto il tasto destro del mouse) e di ascoltare (senza premere nulla). Al termine sarà anche possibile salvare la connessione, in modo da poterla riaprire successivamente; naturalmente questo funzionerà

soltanto in caso di indirizzo IP fisso, cioè tipicamente in caso di connessione dedicata (24 ore su 24). Altre opzioni simpatiche di Speak Freely permettono di inviare dei file in formato .wav o .au con suoni preregistrati, oppure file GIF 128x128, o bitmap non compressi, contenenti una nostra immagine; c'è poi la possibilità di configurare una vera e propria segreteria telefonica (Answering machine dal menù Connection) che registrerà i messaggi in arrivo su hard disk, caso mai dovessimo assentarci dal computer. Ma quello che ci interessa maggiormente, la cifratura della voce, necessita un discorso più approfondito.

Compressione

La compressione della voce è necessaria per non intasare la banda disponibile. Il problema non si pone se telefoniamo in una rete locale ad alta velocità; per l'utilizzo che interessa a noi, però, i pacchetti di dati devono viaggiare attraverso l'enorme rete Internet e possono incontrare colli di bottiglia strettissimi. È quindi necessario operare una compressione della voce digitalizzata, specialmente se poi disponiamo soltanto di un modem a 14.400 bit per secondo (bps) e di una connessione a Internet "lenta". Dal menù Options è possibile selezionare la compressione: *No compression* richiede una banda di circa 8.000 caratteri per secondo (cps), equivalente a circa 65.000 bps, velocità raggiungibile facilmente in rete locale. *Simple Compression* richiede 4.000 cps, velocità raggiungibile con connessioni digitali ISDN. Di default è abilitata la compressione *GSM*, lo stesso algoritmo utilizzato dai più recenti telefoni cellulari; esso riduce ad un quinto le pretese di banda, che risultano essere di circa 1.650 cps: quanto può facilmente trasmettere un modem a 28.800 bps connesso a Internet; da notare che per comprimere i dati con questo algoritmo è necessario un computer relativamente veloce (consigliati un Pentium o almeno un 486 DX2). La compressione *ADPCM* (*Adaptive Differential Pulse Code Modulation*), invece, riduce la banda richiesta a 4.000 cps ed è indicata per quei computer troppo lenti per utilizzare la GSM, ma che dispongono di una banda elevata. È possibile combinare la compressione Simple con la GSM o con la ADPCM senza aggravare le richieste hardware, ma riducendo quelle di banda: Simple + GSM permette di utilizzare un modem 14.400, poiché richiede solo 825 cps, mentre Simple + ADPCM richiede 2.000 cps, gestibili con un modem 28.800. La compressione *LPC* (Linear Predictive Coding) richiede un processore molto veloce dotato di coprocessore matematico e può ridurre i dati di un fattore 12 (650 cps), pur peggiorando in qualità; se abbiamo paragonato la compressione GSM a quella dei telefonini, la LPC può essere accostata a una radio a onde corte:

qualità molto bassa ma, come le radio a onde corte, ci permette di comunicare, e va usata quando nient'altro sembra funzionare. Infine la compressione *LPC-10* è una particolare versione di LPC che permette di comprimere i dati di 26 volte (fino a 346 cps) e anch'essa richiede processori veloci. Nella tabella sottostante sono riassunti i tipi di compressione ora descritti; ricordiamo che ove è richiesto un processore veloce, questo deve operare con poche o addirittura nessun'altra applicazione aperta.

Compressione	CPS richiesti	Richiede una CPU veloce?	Qualità suono
No compression	8.000	No	Ottima
Simple	4.000	No	Povera
ADPCM	4.000	No	Buona
Simple + ADPCM	2.000	No	Scarsa
GSM	1.650	Sì	Buona
Simple + GSM	825	Sì	Rumorosa
LPC	650	Sì	Dipende…
LPC-10	346	Assolutamente sì	Scarsa

Conversazioni cifrate

Speak Freely, al contrario del PGP, *non* è un programma di crittografia a chiave pubblica, bensì a chiave segreta. Ciò significa che gli utenti non avranno la classica coppia di chiavi pubblica e privata, ma concorderanno un'unica chiave segreta, detta *chiave di sessione* (o session key), con la quale crittare la conversazione. Se qualcuno riesce a impossessarsi della nostra chiave di sessione potrà ascoltare la conversazione – o addirittura intervenire a nostra insaputa, come spiegheremo meglio più avanti; concordare la chiave di sessione e tenerla segreta è dunque il passaggio più delicato.

La chiave di sessione è una sequenza di byte (fino a 255). È ovvio che si possono scegliere session key anche di pochi caratteri, come "pippo" o "pluto", ma *non* è la soluzione da adottare se teniamo alla riservatezza della conversazione; più la sequenza di byte sarà casuale, più difficile sarà cercare di ricostruirla a tentativi. Speak Freely può cooperare con il PGP e utilizzare le sue funzioni per generare e scambiare una session key di 128 byte con l'interlocutore: dal menù Options selezionare Connections: nel campo PGP user name inserire il nome dell'utente con cui si vuole parlare – e di cui si possiede la PGP public key – e cliccare su OK. Verrà generata una chiave di sessione (con procedimento casuale) che verrà automaticamente crittata con il PGP e poi altrettanto automaticamente trasmessa all'interlocutore. Speak

Freely può anche generare "manualmente" (menù Options - Create Key) una session key: spetterà poi all'utente farla pervenire al destinatario prima che abbia inizio la conversazione, magari tramite un normale messaggio crittato. Senza aver prima concordato una session key, manualmente o tramite Speak Freely e PGP, è impossibile stabilire una conversazione cifrata, al contrario ad esempio di PGPfone, come vedremo più avanti.

Una volta scelta la session key, Speak Freely può utilizzare tre metodi per cifrare la conversazione: tramite algoritmo DES (Data Encryption Standard) parzialmente semplificato e ormai piuttosto insicuro, tramite algoritmo IDEA (International Data Encryption Algoritm) o tramite key file, o anche tramite combinazioni di questi. I primi due sono algoritmi utilizzati anche da altri programmi di crittografia e sono sufficientemente robusti: lo stesso PGP classico si basa su algoritmi IDEA; viceversa la scelta di un key file, cioè di un file binario il più casuale possibile e di almeno 512 byte, scelto tra quelli che avete sull'hard disk, riduce di gran lunga il carico del processore, ma anche il livello di sicurezza: va vista quindi come ultima risorsa, quando si dispone solo di un computer lento.

PGPfone

PGPfone è, come Speak Freely, un programma per cifrare le conversazioni su Internet; il copyright appartiene a Philip Zimmermann, già noto quale autore del PGP; il programma è distribuito nelle versioni Windows (95/98 e NT, non è previsto supporto per la 3.11) e Macintosh; non è nemmeno previsto supporto per Linux, il sistema operativo freeware più diffuso.

La macchina minima richiesta per installare il PGPfone è un Pentium o un Power Mac dotato di modem che vada a 14.400 bps o più. Come già notato nell'analisi di Speak Freely, un modem a 28.800 o a 33.600 incide notevolmente sulla qualità della conversazione; d'altronde i modelli a 14.400 sono praticamente scomparsi dalla circolazione, come avvenuto qualche anno fa per i modelli inferiori; per quel che riguarda la velocità il processore, PGPfone è più esigente di Speak Freely, anche se non di molto.

La fondamentale caratteristica di PGPfone è il non aver bisogno di un canale sicuro per lo scambio della *session key* prima dell'inizio della conversazione. Le due parti negoziano le loro chiavi usando il protocollo Diffie-

Hellman, che non rivela nulla di utile a un eventuale ascoltatore e permette alle due parti di concordare una chiave comune che useranno per crittare e decrittare le conversazioni.

L'installazione ha meno opzioni di Speak Freely ed è quindi più semplice. La minor quantità di comandi e la loro migliore disposizione logica ne fanno un programma più adatto ai principianti. Ci sono tre tipi di layout, definibili dal menù Window: *minimal* (essenziale), *intermediate* (mostra anche compressione e tipo di cifratura) e *advanced* (con diversi checksum e statistiche). Tutto il resto si setta dal menù Edit - Preferences, che a sua volta è diviso in tre sezioni: Phone, Modem ed Encryption.

Connessione diretta: sezione Modem

Una nota importante: PGPfone, al contrario di Speak Freely, permette di conversare non solo via Internet, ma anche tramite connessione diretta, cioè tramite il classico telefono. Nel caso di connessione Internet non sarà necessario settare la sezione Modem; viceversa, questo sarà il primo passo da affrontare se si intende telefonare direttamente all'interlocutore o ricevere da lui una telefonata. In questo caso vanno qui indicate le impostazioni della porta seriale, la velocità del modem e la stringa di inizializzazione. Quest'ultima non è la banale ATX3 che va bene per la maggior parte dei modem italiani, ma bisognerà indicare che:

1) il modem non deve rispondere in autoanswer quando squilla il telefono (ATS0=0)

2) ogni compressione e correzione d'errore (V42bis o MNP4/5) quale che sia va disabilitata. Per questa opzione occorre consultare il manuale del proprio modem, visto che non esiste un comando standard. Togliere la compressione è importante per migliorare le prestazioni di PGPfone, in quanto una compressione su dati già compressi porterebbe inevitabilmente a un aumento del pacchetto. Analogamente è importante disabilitare la correzione d'errore, dannosa per PGPfone anche se abbiamo delle linee molto disturbate: comporterebbe una ritrasmissione di pacchetti e un "ritardo" nella trasmissione della voce che via via si accumula e diventa percettibile e fastidioso. Solo in caso di linee telefoniche estremamente pulite (strano a dirsi, le linee italiane digitali sono in media decisamente migliori di quelle americane) e quindi quasi esenti da disturbi, può essere conveniente lasciare attiva la correzione d'errore, anche se può sembrare un controsenso. In realtà così facendo si forza il modem a trasmettere in modo sincrono, senza bit di start

né di stop, trasmettendo quindi "parole" di 8 caratteri anziché 10 e guadagnando il 20% di velocità - salvo poi perderla tutta in caso di un piccolo disturbo.

È anche possibile iniziare una conversazione telefonica in maniera tradizionale e successivamente passare in modalità cifrata; nella conversazione in chiaro le due parti devono prima concordare quale dei due modem originerà la connessione (*originate*: comando ATD) e quale dei due sarà il risponditore (*answer*: comando ATA). Solitamente si conviene che chi telefona funga da *originate* anche al momento di passare in cifrato, e chi riceve rimanga *answer*.

Nota: PGPfone non funziona con i cosiddetti *Winmodem*, né con gli Apple Geoport Telecom Adapter: infatti questi tipi di modem, solitamente molto economici, sono costituiti da una componente hardware e da una componente software; quest'ultima demanda alla CPU anche i compiti di un normale modem, sovraccaricandola e rendendo difficoltoso - se non impossibile - l'utilizzo di altro software oltre a quello di comunicazione.

Sezione Phone

La sezione Phone invece permette di configurare parametri quali il nome (non c'è posto per l'indirizzo e-mail e altre informazioni aggiuntive), due tipi di compressione in ordine di preferenza e la scelta tra la connessione diretta modem-modem (cioè tramite una telefonata diretta tra noi e il nostro interlocutore, trattata nel precedente paragrafo) o attraverso la rete Internet. La versione Mac permette anche l'utilizzo in rete locale AppleTalk, mentre per i PC è previsto l'uso all'interno di una LAN solo se è una Intranet, cioè una rete locale in tecnologia TCP/IP. Nel caso di conversazioni via Internet bisognerà conoscere l'indirizzo IP della persona con cui vogliamo parlare - e qui non sono previsti LWL server o altre comodità. Inoltre, al contrario di Speak Freely, non sono permesse più conversazioni simultanee.

L'opzione *incoming calls* permette di mostrare il campo del nome solo quando qualcuno ci chiama; analogamente *outcoming calls* permette di mostrarlo solo alle persone da noi chiamate. Da notare che ciò avviene soltanto a connessione avvenuta; ciò ha il suo peso se non desideriamo far sapere la nostra identità ad altri che il nostro interlocutore. Il box *listen for calls* va lasciato selezionato, a meno che non desideriamo utilizzare un altro software per il modem contemporaneamente a PGPfone.

La scelta tra half duplex e full duplex (attualmente non disponibile nella versione Windows), rimanda alle considerazioni già espresse per Speak Freely; si ricordi solo che, in full duplex, entrambe le parti devono utilizzare le cuffie nonché essere in possesso di scheda sonora dotata di tale capacità.

La scelta della compressione qui si restringe a ADPCM e una decina di varianti di GSM (divise tra GSM propriamente detto e GSM lite, di qualità leggermente inferiore ma meno gravosa per la trasmissione), ognuna con qualità audio differenti.

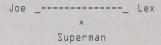
Sezione Encryption

Sono disponibili tre algoritmi di cifratura: Blowfish (il migliore in termini di rapporto sicurezza/prestazioni), Triple DES (simile al DES di Speak Freely: molto buono, però richiede hardware veloce), o CAST (un compromesso tra i due). Non è invece disponibile l'algoritmo IDEA, utilizzato nel PGP tradizionale: questo a causa di una ditta svizzera, la Ascom-Systec, che detiene il brevetto di quest'algoritmo. Dopo che il PGP ha reso famosa la cifratura con algoritmo IDEA, la Ascom ha aumentato i prezzi per le licenze software del proprio algoritmo: oggi, in Europa, la Ascom chiede circa un centinaio di dollari a ciascun utente commerciale di PGP per poter utilizzare l'algoritmo brevettato.

L'uomo nel mezzo (Man-in-the-middle-attack)

Abbiamo accennato poco sopra che usando PGPfone la chiave di sessione con cui cifrare la conversazione non viene concordata anticipatamente attraverso un canale sicuro, bensì viene concordata tra le due parti attraverso PGPfone stesso. Il problema maggiore che potrebbe capitare è il cosiddetto "attacco dell'uomo nel mezzo". Questo non è un problema specifico dei nostri giorni, ma è vecchio di migliaia di anni. Immaginiamo due commercianti, uno in Italia e uno in Spagna, mai incontrati di persona, che vogliono comunicare per posta. Lo spagnolo fa una proposta di vendita all'italiano e firma la lettera. Come può l'italiano sapere se la firma è vera, se non ha mai incontrato prima lo spagnolo? Un impiegato postale in Francia potrebbe intercettare la *prima* lettera tra Spagna e Italia, riscriverla e firmarla lui stesso, fingendo di essere il commerciante spagnolo. Se l'italiano cade nel tranello,

l'impiegato francese potrà intercettare tutte le lettere e i contratti successivi, riscrivendoli e continuando a spacciarsi per lo spagnolo.

La versione moderna dell'attacco dell'uomo nel mezzo è molto simile. Supponiamo che le due parti, Joe Lametta e Lex Luthor, vogliano aprire una conversazione privata con PGPfone; supponiamo che ci sia una terza persona, Superman, che voglia ascoltare la conversazione, e che si interponga tra Joe e Lex, spacciandosi con Lex per Joe e analogamente con Joe per Lex.

Al momento di concordare la chiave di sessione, se è presente l'uomo nel mezzo (Superman), non saranno più semplicemente Joe e Lex a concordare la chiave. A loro insaputa verranno concordate due chiavi, quella di ciascun interlocutore con Superman: una prima tra Joe e Superman, che con Joe si finge Lex; una seconda tra Lex e Superman, che con Lex si finge Joe. Al momento della conversazione vera e propria, Joe parlerà a Superman utilizzando la prima chiave di sessione; Superman ritrasmetterà a Lex utilizzando la seconda chiave. Lex per conto suo non si accorgerà dell'uomo nel mezzo e penserà di aver parlato con Joe; la stessa cosa succederà all'incontrario nella conversazione che da Lex va verso Joe. In pratica ci sono *due* canali cifrati con *due* chiavi differenti. La cosa è tecnicamente possibile e il problema è serio: le due parti (Joe Lametta e Lex Luthor) sanno di aver stabilito una connessione "sicura", ma non sono sicure di averla stabilita effettivamente tra di loro. Se poi Superman utilizza un computer per registrare quel che Joe dice e ritrasmetterlo in tempo reale a Lex e viceversa, ci sono buone probabilità che i due non si accorgano che c'è Superman a intercettarli, visto che la conversazione non subisce rallentamenti percettibili.

Firma biometrica

Per fare fronte al man-in-the-middle-attack, il team di Phil Zimmermann ha utilizzato una buona contromisura: Joe Lametta e Lex Luthor dovranno verbalmente comunicarsi l'un l'altro qual è la chiave di sessione che stanno utilizzando. Per farlo dovranno, all'inizio di una conversazione cifrata, leggersi rispettivamente la *firma biometrica*, ovvero una serie di parole (*biometric signature*) che compariranno in un'apposita finestra di PGPfone, uguali per entrambi e che identificano univocamente la chiave di sessione utilizzata. Se non c'è alcun uomo-nel-mezzo, Joe leggerà la lista di parole a Lex, il quale

vedrà che corrisponde alle parole che compaiono sul suo monitor e la conversazione sarà effettivamente privata. In caso di presenza del terzo uomo, le cose saranno differenti: Joe leggerà la sua lista di parole a Superman; se Superman rileggesse quella stessa lista a Lex, Lex si accorgerebbe che non è la lista che ha sul suo monitor: infatti Joe e Superman utilizzano una chiave differente di quella utilizzata tra Superman e Lex e anche la lista di parole sarà differente. In questo caso, accorgendosi che le liste non coincidono, Lex capirà che c'è qualcuno che sta intercettando la conversazione e agirà di conseguenza: potrà rimandare la conversazione, oppure far finta di niente e non parlare di nulla di riservato.

La *biometric signature* è qualcosa di simile all'alfabeto militare, che comprende 26 parole di uso internazionale come *alfa, india, tango* ecc. Ciascuna parola corrisponde a una lettera dell'alfabeto. PGPfone estrae la biometric signature da un elenco di 256 parole anziché 26, ciascuna accuratamente scelta e foneticamente ben distinguibile dalle altre (almeno per gli anglofoni). Le primissime versioni di PGPfone prevedevano una lettura di una *fingerprint PGP* lunga 16 bytes; ma leggere una sequenza di 16 bytes in esadecimale può generare molti più errori o incomprensioni che leggere una lista di parole: così si è deciso di usare una lista di parole in cui ogni parola rappresenta un byte.

Si potrebbe obiettare che il Diffie-Hellman, cioè l'algoritmo usato in PGPfone per la negoziazione della chiave attraverso un canale non sicuro, non sia altrettanto sicuro dell'aver stabilito a priori la chiave di sessione ed essersela scambiata usando un canale non passibile di intercettazione, quale ad esempio un messaggio e-mail crittato con PGP. Non è così. Se è vero, infatti, che la negoziazione avviene "pubblicamente", cioè agli occhi di un potenziale nemico, questi non ricava alcuna informazione utile a ricavare la chiave di sessione dai dati che le due parti si scambiano pubblicamente. Senza scendere in dettagli, che peraltro possono essere approfonditi consultando un buon testo di crittografia, basti pensare che Joe Lametta e Lex Luthor condividono in pubblico solo *una parte* delle informazioni necessarie alla costruzione della session key; Joe, utilizzando le informazioni che gli provengono da Lex dal canale non sicuro, e computandole con altre informazioni che non ha trasmesso a Lex – e che quindi conosce solo lui – giunge a stabilire una chiave di sessione. Nello stesso tempo Lex, basandosi sui dati ricevuti da Joe "pubblicamente", e confrontandoli con altre informazioni che *non* ha trasmesso a Joe, riesce ad arrivare alla *medesima* chiave di sessione (ciò è possibile grazie a complessi calcoli matematici che non riportiamo qui, ma che sono chiaramente indicati nel manuale di PGPfone). L'intercettatore Superman, che ha a disposizione *solo* dati circolati in "pubblico", non arriverà mai alla chiave di sessione: infatti i dati circolati in pubblico

sono sì correlati ai dati che Joe e Lex non si sono trasmessi, ma non è possibile – a meno di anni e anni di lavoro-macchina – risalire a questi ultimi e quindi ricostruire la chiave di sessione; nello stesso tempo i dati circolati in pubblico, combinati con quelli che Joe e Lex non si sono trasmessi, permettono ad essi – e *solo* ad essi – di giungere alla medesima chiave di sessione, che l'intercettatore Superman non conoscerà mai.

Superman non si arrende: Rich Little Attack

Posto che l'unico attacco cui è vulnerabile l'algoritmo Diffie-Hellman è quello del terzo uomo, come può comportarsi l'intercettatore Superman per mettere a segno tale attacco senza farsi scoprire? Potrebbe imitare la voce di Joe Lametta a Lex Luthor e viceversa, ovvero operare un "Rich Little Attack". Al momento della lettura delle chiavi, potrebbe leggere a Joe la lista di parole della chiave Joe-Superman e a Lex la lista della chiave Superman-Lex. In questo modo le liste di parole corrisponderebbero sia per Joe, sia per Lex, pur essendo differenti: Joe e Lex in questo caso sarebbero convinti di parlare solo fra di loro. Affinché questo possa succedere, però, Superman dovrebbe continuare a imitare la voce di Lex a Joe e viceversa per tutto il resto della conversazione: altrimenti i due, riconoscendo il cambiamento di voce, capirebbero che qualcuno si è intromesso al momento della lettura della firma biometrica. Chiaramente questo procedimento può funzionare solo in quei casi in cui Lex e Joe non si conoscono di persona, o meglio, non conoscono ciascuno la voce dell'altro. In questo caso, l'attacco dell'uomo nel mezzo potrebbe funzionare. Da un lato ciò dimostra una implicita debolezza del PGPfone, dall'altro bisogna tener conto che è estremamente improbabile che ci sia qualcuno così abile nel mascherare *in tempo reale* la propria voce imitando quella di altre due persone che (presumibilmente) si conoscono. Gli attacchi-dell'uomo-nel-mezzo, soprattutto, sono attacchi *rischiosi* (in gergo vengono definiti *danger attacks*, contrapposti ai *safe attacks* tradizionali), nel senso che è probabile che vengano scoperti.

Se però Joe Lametta e Lex Luthor non si conoscono, o prendono l'abitudine di *non* leggere quella noiosa lista di parole all'inizio della conversazione, e se Superman lo viene a sapere (o lo immagina), allora potrebbe operare davvero un attacco dell'uomo nel mezzo, e in questo caso l'attacco *funzionerebbe*. Lasciamo il secondo caso agli imprudenti e consideriamo solo quello in cui due persone non conoscono sufficientemente ciascuno la voce dell'altro.

In questo caso, purtroppo, il PGPfone è potenzialmente vulnerabile. Certo, è estremamente improbabile che un attacco del tipo sopra descritto venga portato a segno con successo, ma non impossibile. Inoltre non è vero che due persone debbano necessariamente conoscersi di persona per avere qualcosa di privato da condividere, soprattutto oggi che ci si incontra e conosce nelle reti telematiche e su Internet; basti pensare che persino gli autori di questo libro, pur lavorando in team, non si conoscono tutti di persona.

E se Superman registra tutto?

Non è necessario che Superman agisca in tempo reale: potrebbe benissimo intercettare l'intera conversazione cifrata e registrarla, per poi tentare una decodifica in seguito, con calma e senza timore di essere scoperto. Non essendo in possesso della chiave di sessione che, ripetiamo, *non è mai circolata* nel canale pubblico ma è stata calcolata dalle parti tramite uno scambio di dati nel canale pubblico, Superman teoricamente impiegherà molti anni-macchina prima di riuscire decifrare il messaggio. Non dimentichiamo però che: la tecnologia migliora di giorno in giorno: se, per esempio, oggi occorrono 300 anni per decifrare un messaggio, tra un anno probabilmente ne occorreranno "solo" 100 o 50; sfruttando in parallelo la potenza di calcolo di decine (o centinaia) di macchine, non necessariamente localizzate nello stesso posto ma in una qualsiasi parte del globo, è possibile ridurre i tempi di calcolo a un decimo (o a un centesimo) del tempo attuale.

Cavallo di Troia

Un altro possibile attacco è quello del Cavallo di Troia (ne abbiamo già parlato nel capitolo sulla crittografia). Supponiamo che Superman metta in giro (inviandole in siti FTP o facendole comunque pervenire a Joe Lametta e a Lex Luthor) versioni di PGPfone appositamente "truccate", cioè modificate in modo da permettere a un terzo (nello specifico a lui stesso) l'ascolto della conversazione o la sua decodifica in un secondo tempo. Per evitare questo e per far sì che Joe e Lex siano sicuri che la loro copia di PGPfone sia effettivamente priva di backdoor, devono assicurarsi che questa sia "garantita" da Zimmermann e soci tramite la PGP signature, ovvero quella firma elettronica che assicura che è stato Zimmermann (o persona di sua fiducia) a rilasciarla.

Zimmermann afferma che un problema simile si è già verificato con lo stesso PGP: essendo i sorgenti pubblici, ne sono nate parecchie versioni cosiddette "bogus", ovvero modificate da terzi e potenzialmente non prive di backdoor. Questo discorso potrebbe sembrare una operazione commerciale: "Non fidatevi di altro PGPfone se non di quello che comprate da me" ma considerando che almeno finora tutti i prodotti di Zimmermann sono sempre stati rilasciati anche (o soltanto) in versione gratuita, è un discorso che regge poco; non gli si può tuttavia negare un fondo di verità. A questo problema se ne aggiunge un altro: quello dell'esportazione dagli Stati Uniti, che rende impossibile per un utente europeo il prelievo della sua copia di PGPfone direttamente dal sito "ufficiale" del MIT. Esistono comunque siti europei che ormai sono praticamente "ufficiali", in quanto mettono a disposizione in tempi brevissimi tutto il software relativo al PGP e suoi derivati.

Nautilus

Un altro programma di telefonia digitale crittata è Nautilus, il cui nome viene dal noto romanzo di Jules Verne, *20.000 Leghe Sotto I Mari*; mentre il sottomarino del Capitano Nemo affondava le navi, il nostro Nautilus è nato per "affondare" il progetto del Clipper Chip, lo standard nazionale di cifratura proposto a partire dal '93 da Clinton, largamente osteggiato poiché lascia al Governo USA la possibilità di decifrare le conversazioni.

Nautilus è gratuito e, analogamente agli altri programmi trattati, i suoi sorgenti sono disponibili; può girare su computer molto vecchi (la richiesta minima è un 386 a 25Mhz) e questo è senz'altro un punto a suo favore; se è vero che pochi di noi avranno sulla scrivania un computer così datato, è altrettanto vero che sicuramente ci sono parecchi computer *portatili* di classe 386 o 486 ancora in utilizzo; e un computer portatile ha la interessante caratteristica di poter essere utilizzato con un telefono cellulare o anche in una cabina telefonica con la presa dati, ove disponibile (se ne vedono a volte negli USA, non ancora in Italia). Altro fattore da non trascurare è che a Nautilus può bastare una connessione modem a bassa velocità: 14.400 bps sono più che sufficienti, ma il programma può operare anche a 9.600 o a 7.200; la versione provata (1.5) dispone anche di un protocollo di compressione avanzato che necessita di una connessione a soli 4.800 bps, il che è ottimo soprattutto nell'ambito di una connessione cellulare, in genere disturbata e a bassa velocità. Chiaramente valgono tutte le considerazioni già espres-

se per PGPfone e Speak Freely: una maggiore compressione chiede meno banda, ma necessita di una CPU più veloce e la qualità della voce ne risente.

Nautilus può girare su PC (con MS-DOS, Linux o Solaris X86) o su una Sun Sparcstation (con SunOS o Solaris); l'utilizzo con sistemi operativi quali Windows o DosEmu di Linux è sconsigliato: non solo per motivi di sicurezza – la session key potrebbe rimanere all'interno dello swap file – ma anche per effettivi problemi di multitasking. Gli autori inoltre sconsigliano vivamente l'utilizzo di Nautilus in una finestra DOS di Windows 95/98, perché "potrebbe causare seri danni al file system", in verità da noi non riscontrati; invece ci siamo imbattuti in specifici problemi con Windows NT.

Il funzionamento di Nautilus è simile ai due programmi già analizzati più indietro; la differenza fondamentale è che – almeno nella versione DOS – non è possibile effettuare la connessione attraverso la rete Internet ma è necessario telefonare direttamente; uno dei due modem si metterà in attesa (modalità *answer*) e aspetterà la chiamata dell'altro (*originate*), ovviamente stabilita prima tra le due parti. A connessione avvenuta avviene la negoziazione della session key tramite l'algoritmo Diffie-Hellman e la lettura a voce della chiave biometrica, procedura già ampiamente analizzata nella parte sul PGPfone; a differenza di questo, però, è anche possibile stabilire prima della connessione – e possibilmente attraverso un canale sicuro, quale ad esempio un messaggio crittato – una passphrase: così l'inconveniente del man-in-the-middle-attack viene scongiurato definitivamente. La connessione diretta implica una maggiore velocità, che rende la conversazione certamente molto più intelligibile di una effettuata via Internet: non è soggetta a frazionamenti tra una parola e l'altra ed evita fastidiose ripetizioni di parole o di frasi non comprese. Il rovescio della medaglia è la tariffa da pagare, che non è quella locale di accesso a Internet, ma quella telefonica dipendente dalla distanza tra i due interlocutori e quindi generalmente più elevata.

Anche in Nautilus è possibile scegliere tra diversi algoritmi di compressione e tra diversi metodi di cifratura: Triple-DES, Blowfish (pressoché identici a quelli utilizzati in PGPfone) e IDEA (già utilizzato in Speak Freely, nonché in PGP). La conversazione avviene soltanto in half-duplex: è necessario premere un tasto per passare dalla modalità "parla" alla modalità "ascolta"; poiché la connessione è diretta e non si passa attraverso Internet, non ci sono problemi di ritardo tra un pacchetto e l'altro: in questo caso il full-duplex sarebbe stato davvero utile. Non si può che attendere un suo implemento in una versione futura, o prendere in mano i sorgenti e programmarla di persona...

Nautilus è un prodotto che non ha nulla da invidiare, in quanto a sicurezza e funzionalità, agli altri programmi trattati; purtroppo però non è per nulla intuitivo da usare e questo potrebbe essere uno scoglio insormontabile per l'utente medio. Tutti i comandi vanno dati da command-line, non c'è interfaccia grafica e – purtroppo – spesso per ottenere una connessione è necessario resettare il modem o addirittura l'intera macchina. Per chi ha lavorato su DOS e per i linuxisti non dovrebbero esserci problemi, viceversa chi conosce solo Windows è meglio che si orienti su qualcos'altro, o chieda a qualche amico programmatore di costruirgli un front-end opportunamente semplificato.

Sebbene gli autori di Nautilus abbiano in cantiere parecchie implementazioni future, bisogna notare che il programma non viene più aggiornato da tempo e questo certo non è un bene perché comincia a risentire dell'età.

Conclusioni

Dopo aver analizzato questi tre programmi, ognuno con pregi e difetti, non è facile consigliarne uno rispetto a un altro. Chi è più tecnicamente ferrato preferirà Nautilus, che è più efficiente e veloce; chi ha una connessione veloce ad Internet sicuramente si orienterà su PGPfone o Speak Freely, a seconda che privilegi la facilità d'uso o desideri la sicurezza più assoluta.

Chi non fosse interessato alla cifratura della voce, ma soltanto a farsi una chiacchierata via internet con un amico lontano, può utilizzare decine di programmi di telefonia non crittata reperibili facilmente in rete; viceversa, chi avesse la necessità di scambiare informazioni velocemente e al riparo da occhi indiscreti, senza ricorrere a software di telefonia, può molto più semplicemente ricorrere alla classica accoppiata e-mail + PGP. Chi fosse totalmente ostile all'idea di utilizzare un computer e decidesse quindi di rinunciare a una buona fetta di sicurezza, nonché a qualche litro di sangue, potrebbe optare per lo scrambler della ComSec, a patto di riuscire ad esportarlo (illegalmente) dagli Stati Uniti. Sicuramente il gioco non varrebbe la candela: tanto vale parlare al telefono in chiaro.

Al di là di questo, una piccola riflessione. Non sempre è necessaria la crittanalisi per scoprire un segreto; spesso sono sufficienti altri metodi, meno costosi in termini di risorse e di tempo e quindi più facilmente praticabili. Probabilmente la crittonalisi è *l'ultimo* tentativo che un nemico potrebbe tentare nei nostri confronti. Ad esempio piazzare una microspia è decisa-

mente più "facile" ed economico che non tentare di attaccare direttamente gli algoritmi di crittografia. Analogamente è più semplice corrompere qualcuno che ci sta vicino affinché riveli le nostre informazioni confidenziali, o ancora – anche se sembra fantascienza – qualcuno può puntare un laser contro la nostra finestra per captarne le vibrazioni generate dalla nostra voce e ricostruire in questo modo le nostre conversazioni. Si può anche utilizzare un apparecchio (posizionato nelle vicinanze) che capta i segnali elettromagnetici generati dal nostro computer (*Van Eck Attack*): questo è un metodo costoso ma riutilizzabile in mille altre situazioni e quindi sempre più economico che impiegare qualche cervellone a tentare una crittanalisi vera e propria. Di tutto questo si è parlato nel paragrafo sugli "attacchi pratici" del capitolo sulla crittografia. Perché aggiungere ulteriori paranoie? Perché nonostante ci si possa dotare di tutto l'armamentario crittografico esistente al mondo, esso non è sufficiente - sempre e comunque - per essere del tutto tranquilli. Più che un'arma infallibile, ogni programma di crittografia va considerato solo come *un'arma a disposizione in più.*

Bella storia, questa del telefono. E noi che per anni ci siamo fatti mille paranoie ogni volta che dovevamo dirci qualcosa.

Il mio povero nonno mi aveva insegnato un bel po' di parole in codice, per indicarmi cosa gli dovevamo portare io e la nonna quando andavamo a fargli la visita in galera. Pace all'anima sua, chissà cosa penserebbe di queste cose moderne. Nel frattempo amico, Superman cominciava a dar fuori di matto. Quando ha capito quello che stava succedendo è rimasto per un bel pezzo seduto senza dire una parola davanti ai suoi schermi, lì nella Fortezza della Solitudine, come la chiama lui. Poi si è alzato in piedi e si è guardato allo specchio. Non è mica uno troppo abituato a pensare, il nostro SuperBamboccione. Ma quello che gli è passato per la testa te lo puoi immaginare anche tu, amico. Battuto per la prima volta nella sua vita, da un nanerottolo orrido come il qui presente. Con uno splendido futuro da SuperVuotacessi di fronte. E con un paio di mutandine sporche come unico ricordo della sua donna. Le palle hanno cominciato a fumargli talmente che lo specchio davanti a lui rischiava di fondersi. Ha lanciato un urlo disumano che si è sentito sino in Giappone, tanto che i gialli per un po' hanno pensato che il vecchio Godzilla stesse tornando a dare un'altra ripassatina a Tokyo e hanno messo in allarme l'esercito. Poi è decollato. Stavolta niente buchi nelle pareti, amico. Una voragine. Sembrava il lancio del Saturn 3. E una volta in volo ha cominciato a sterminare tutte le dorsali di Internet, sissignore. Individuava i cavi con i SuperSensi e li bruciava con i raggi calorifici. Senza stare a guardare troppo per il sottile se magari nei paraggi c'era qualche raffineria o qualche impianto nucleare. Per un po' sembrava che fosse scoppiata davvero la Guerra Dei Mondi, come in quel vecchio film. Te l'immagini, amico? Raggi verdi che piovevano all'improvviso dal cielo ed esplosioni dappertutto. Ce l'hai presente il cavone sottomarino, la dorsale principale che passa sul fondo dell'Oceano? Beh, amico. Il nostro SuperEroe si è tuffato a pesce con tale velocità che intorno l'acqua si è messa a bollire come un pentolone per un raggio di centinaia di miglia. Ha raggiunto il cavo in un battibaleno e l'ha troncato con i denti. Seeeeeh, li ho letti anch'io i giornali, come no: "Inspiegabile serie di incidenti industriali mette temporaneamente fuori uso Internet su tutto il pianeta". Tutte palle. Le cose stanno come ti ho detto, e anche se non ci credi chissenefotte. Ok, facciamoci portare un'altra bottiglia, te l'ho detto che dopo la terza

sembra buono, eh? Beh, per farla breve, i capoccioni di tutto il mondo cominciavano a essere preoccupati. Anche perché gli scherzetti di Superman a quel punto gli stavano costando più morti (e soprattutto più dollari) di quel che poteva costargli il piccolo affaruccio che avevano in corso con noi. Per cui hanno pensato bene di mandarlo a chiamare per vedere di calmarlo un po'. L'incontro segreto è avvenuto all'ONU, perché il vecchio Bill dopo la prima esperienza ha pensato bene di rifilare a loro la SuperPatata Bollente. Superman se ne è restato lì per un bel pezzo, mentre gli spiegavano che ultimamente sembrava un po' troppo sotto pressione, che apprezzavano i suoi sforzi ma la sua salute era preziosa per il mondo e forse faceva meglio a prendersi una settimana di riposo, che gli industriali, gli ambientalisti, i militari, i fanatici di Internet e tutti quanti erano preoccupati e bla bla bla. Morale della favola: si mettesse a coltivar lattuga e lasciasse perdere la questione della Bomba a Metropolis. L'Omino d'Acciaio è un tizio di poche parole, questo almeno bisogna riconoscerlo. Li ha lasciati parlare sino a seccarsi la lingua e quando hanno finito ha detto solo: "NO". Ma l'ha detto in modo tale che a quel punto tutti hanno capito che non era il caso di insistere. Tutti tranne uno. Beh, bisogna anche capirlo quel povero Bill Gates. A lui il blocco di Internet e tutto quel casino avevano fatto perdere ben più di un miliardo di dollari, poveretto. Era già incazzato nero da tempo, in particolare per la storia della crittografia fuorilegge. Lui sulla crittografia ci campava. Quando costruisci il tuo impero vendendo software e altre informazioni, non c'è niente di meglio di un buon sistema di crittografia per pararti il culo da spioni, concorrenti, dipendenti malpagati e compagnia bella. Si era fatto dare una poltrona all'ONU apposta per difendere la crittografia dalle zampe dei giudici e delle leggi. "Irrinunciabile diritto dei popoli". "Tutela della dignità e dell'intimità della persona umana"... sticazzi. Senza crittografia anche la gente come lui era col culo per terra. Pronti a essere inchiappettati dal primo hacker da strapazzo noleggiato da chissacchì per farsi una passeggiata sui loro sistemi informatici. Quindi a quel punto non ci ha visto più. Si è alzato e ha cominciato a sbraitare: "Tu, fottuto ammasso di muscoli in calzamaglia senza un briciolo di cervello. Stiamo parlando di FATTURATI qui, la vuoi capire? Di MILIARDI, non delle tue puttanate da eroe dei fumetti da mezzo dollaro. Stampati in quella tua testaccia di cazzo che noi ti ORDINIAMO..." Non ha mica fatto in tempo a finire. Si è beccato una sventola tale che è volato fuori dal Palazzo di Vetro direttamente dal novantacinquesimo piano. Fine della carriera di un genio dell'imprenditoria informatica. Non ne è rimasto un granché sul marciapiede e anche quel poco non era per niente bello a vedersi, non so se mi spiego. Naaaaa, mica è vero che si è suicidato dallo schifo mentre stava guardando il codice di Windows 1999. Amico, te l'ho detto e te lo ripeto, i giornali contano balle. Il vecchio Superman non si è nemmeno affacciato alla finestra per vedere che fine avesse fatto. Ha preso su ed è sparito all'orizzonte. Stavolta l'aveva fatta davvero grossa, ma Lex Luthor e io per la prima volta in questa storia potevamo davvero trovarci con l'acqua alla gola. Mancava pochissimo allo scadere dell'ultimatum, e dovevamo metterci d'accordo su cosa fare se all'ultimo momento il sindaco di Metropolis cercava di farci qualche scherzetto, anche se ormai sembrava pronto a pagare. Con i cavi principali tagliati e i satelliti inceneriti, anche PGP e remailer vari servivano a ben poco. Se solo fossimo riusciti a risolvere anche questo problema…

Packet Radio

di And Bov

Nei capitoli precedenti abbiamo affidato le nostre informazioni a reti cablate accessibili mediante il nostro modem casalingo, che utilizza lunghe distese di doppino telefonico o di fibra ottica per permetterci di comunicare con i nostri interlocutori. Vi sono situazioni che richiedono invece l'abbandono di ogni connessione fisica con la rete: la necessità di connettersi in movimento (ad esempio da una macchina), il trovarsi in zone dove non c'è la disponibilità della rete telefonica, l'impossibilità di cablare e quindi collegare fra loro varie entità senza dover intervenire fisicamente sulle strutture che le contengono come muri o pavimenti. Questi sono solo alcuni dei tantissimi esempi in cui abbiamo bisogno di affidare i nostri byte a un mezzo di trasmissione diverso dal solito doppino o dal cavo coassiale.

L'etere si presta ad essere il mezzo trasmissivo che risolve le situazioni sopra elencate. Fra i tanti metodi di sfruttamento, quello che ci permette di trasmettere *via radio* informazioni digitali è il *packet radio*. Il termine *packet radio* nasce nel 1965 quando viene per la prima volta ipotizzata la trasmissione di pacchetti di dati su un canale radiofonico. Lo scopo era di collegare alcuni tratti di quella rete militare che sarebbe diventata dopo qualche anno Arpanet, il progetto in cui è stato sperimentato e applicato per la prima volta il protocollo TCP/IP che è alla base dell'odierna Internet. Solo nel 1978 il packet radio è stato utilizzato in ambito civile, per la precisione da radioamatori canadesi, con la conversione di un modem a 2400 baud per l'uso su canali radio.

Nelle prossime pagine verranno descritti sia i metodi più semplici e più sicuri per affidare all'etere i nostri dati, sia le proposte commerciali di un settore che nei prossimi anni rivoluzionerà completamente il mondo della telematica.

Come si è detto, nel 1978 un gruppo di radioamatori canadesi interfaccia per la prima volta un modem telefonico a una radio con l'intento di riuscire a ricevere e trasmettere dati attraverso l'etere. Le comunicazioni digitali via radio hanno una storia lunghissima, basti pensare che il *morse* non è altro che un sistema binario basato su due sole transizioni possibili: all'assenza di segnale (o della portante) corrisponde lo zero e alla presenza dello stesso corrisponde l'uno logico. Tuttavia si possono riscontrare varie difficoltà a trasmettere via radio il segnale che siamo abituati ad affidare alle linee telefoniche.

Il primo problema è di solito la *qualità* del segnale trasmesso e, in diretto collegamento a essa, la successiva qualità della ricezione. Bisogna tenere presente che la ricezione avviene in genere a chilometri di distanza, disturbata per di più dalla naturale attenuazione dovuta alla propagazione del segnale radio. Questo fa sì che per i modem radio siano stati attivati dei protocolli che oltre alla normale correzione di errore implementano anche una possibile ricostruzione del segnale in ricezione, permettendo così di raggiungere lunghe distanze.

Un secondo fattore che modifica completamente il rapporto tra la distanza raggiungibile e la velocità dei dati è la frequenza che viene utilizzata per mettere in contatto due stazioni packet radio. Nella Tabella 1 analizziamo le varie possibilità offerte da questo sistema di comunicazione.

Come si può vedere il rapporto velocità - distanza è alla base del funzionamento del packet radio. Con questo sistema possiamo per esempio met-

Frequenza	**Distanza**	**Velocità**	**Tipo di utilizzo**
Hf 3 - 30 Mhz	Migliaia di Km	300 - 1200 Baud	Collegamenti a lunghissima distanza per scambio messaggi
Vhf 120 - 170 Mhz	300 Km	1200 - 19200 Baud	Creazione di reti con accessi fissi e in movimento
Uhf 400 - 500 MHz (in vista ottica)	200 Km	9600 - 38400 Baud	Collegamenti urbani ad alta velocità
Shf 1.2 - 10 Ghz (in vista ottica)	10 - 100 Km	38400 Baud - 1.2 Mbaud	Collegamenti Punto Punto

Tabella 1

tere in collegamento stazioni di diversi continenti attraverso le onde corte (HF) rigorosamente a basse velocità, oppure possiamo creare velocissime *back-bone* via radio, utilizzando però solo collegamenti in vista ottica con parabole per frequenze Uhf e Shf.

La parte non digitale del sistema, rappresentata da radio e antenne, sarà progettata a seconda della frequenza e quindi della portata del proprio sistema. Di solito rappresenta una fase estremamente importante per una buona trasmissione dei dati via radio: la bontà del collegamento radio influisce molto sulla velocità e sulla qualità della trasmissione. È quindi molto importante, al momento della scelta della frequenza e dell'eventuale portata del sistema, sovradimensionare potenze e antenne in modo da essere sicuri di avere un ottimo segnale in ricezione anche da postazioni mobili.

Basse velocità e libertà di movimento

Verso la fine degli anni ottanta un gruppo di radioamatori tedeschi, stanchi di dover costruire *radiomodem* sempre più complessi, decise di avviare un progetto che avrebbe rivoluzionato interamente il packet radio. Realizzarono un software che delegava al computer tutta la fase di implementazione del protocollo e di correzione di errore. Attraverso una semplice scheda dotata di pochi componenti, connessa sulla porta seriale del proprio personal computer, era così possibile modulare in toni audio i dati digitali da trasmettere. Il progetto, chiamato *sistema Baycom*, ha permesso a migliaia di appassionati la sperimentazione del packet radio con una minima spesa e con pochissime conoscenze di elettronica e di hardware.

Il cuore di questo sistema è un programma, progettato per DOS e attualmente disponibile come prodotto shareware anche per Linux, che permette di realizzare sul proprio computer tutte le fasi di compressione, controllo e implementazione del protocollo packet. In precedenza tali fasi venivano svolte da modem radio dotati di microprocessore Z80 con 32kb di memoria RAM e di una primitiva *eprom* contenente il software. Ora, con il Baycom e un piccolo modem da collegare alla seriale, è possibile connettersi a una normale ricetrasmittente per lanciare segnali digitali nell'etere.

Il modem necessario, oltre ad avere un costo ridottissimo (è basato sull'integrato TMC 3105: praticamente un modem in miniatura) è anche molto semplice da realizzare e permette di poter raggiungere subito velocità di 1200-2400 Baud con la maggior parte delle radio Vhf e Uhf in commercio.

Il software, Baycom, permette sia la gestione del protocollo packet radio nativo, (l'AX25, una implementazione amatoriale del noto X.25) sia di trasmettere dati in TCP/IP o in qualsiasi altro protocollo di comunicazione. Baycom è distribuito praticamente ovunque; le ultime versioni per Linux contengono all'interno il driver che permette di gestire la porta radio come una qualsiasi risorsa di comunicazione (praticamente una Ethernet via radio).

Sono anche disponibili dei Baycom modem a 9600 baud che permettono di realizzare piccole reti mobili o fisse in maniera veloce e trasparente, adattandosi ai protocolli di comunicazione più diffusi. Il progetto Baycom mette a disposizione su Internet un sito completo di informazioni, da dove è possibile scaricare il software e gli schemi di costruzione dei modem, oppure si possono acquistare i prodotti montati direttamente dal team tedesco di sviluppatori. Informazione precise e aggiornate, su questa come sulle altre risorse in rete citate in questo libro, sono disponibili alla pagina web di Kriptonite, all'indirizzo:

`http://www.ecn.org/kriptonite`

Oltre il Ghz ad alta velocità

Attraverso il sistema Baycom possiamo risolvere comodamente problemi di trasmissione dati tra unità in movimento o mettere in piedi in poche ore reti di comunicazione per gli usi più vari. Quello che invece non possiamo fare è sostituire i più veloci collegamenti via cavo che connettono le macchine tra loro e permettono l'accesso alle reti telematiche. La soluzione a questo problema arriva attraverso l'uso di sistemi packet che lavorano a frequenze maggiori dove l'uso di ampie larghezze di banda (non realizzabili tecnicamente in Hf e Vhf) permette il raggiungimento di velocità non inferiori a quelle raggiungibili via cavo.

Le frequenze che permettono di raggiungere queste velocità sono le *microonde*, un genere di onda noto per la caratteristica di riscaldare in fretta le cellule di quasi tutti i materiali all'interno dei forni casalinghi. Meno conosciuta è la capacità di queste frequenze di comportarsi come veri e propri cavi per la trasmissioni dei dati, con la sola limitazione che la stazione di trasmissione e quella di ricezione devono essere rigorosamente in portata ottica.

Per sistema di comunicazione in *portata ottica* si intende un sistema(ad esempio, appunto, in microonde) in cui le parabole sono puntate in modo che una linea d'aria immaginaria colleghi senza interruzioni la stazione che trasmette a quella che riceve (sul campo di frequenze delle microonde le antenne più utilizzate sono di forma parabolica o elicoidale). La portata ottica è una limitazione che costringe a progettare e realizzare questo tipo di trasmissioni solo per collegamenti punto a punto e non per un uso circolare che permetterebbe a più utenti di condividere un canale radio.

Le velocità realizzabili su queste frequenze sono decisamente molto interessanti. A livello sperimentale alcuni radioamatori hanno implementato un sistema che raggiunge gli 1.2 Megabaud di velocità sulla frequenza di 1.2 Ghz. A livello commerciale esistono invece vari prodotti che permettono di eguagliare la resa di una linea dedicata (CDN) su distanze di decine di chilometri. Nonostante la praticità e anche l'economicità (sempre a livello amatoriale) di un sistema di comunicazione di questo tipo, esso non può essere automaticamente considerato l'alternativa alla connessione via cavo. I motivi sono vari, sintetizzati nella Tabella 2.

	Costi	**Affidabilità impianto**	**Sicurezza**
Sistema amatoriale	Relativamente bassi per la disponibilità nel settore surplus (recupero di materiale militare) di attrezzatura per microonde (trasmettitori, parabole)	Buona — eventuali problemi di degradazione delle qualità del sistema (banda passante) in caso di intemperie (neve, ghiaccio, forte pioggia)	Media — questi progetti utilizzano modulazioni standard facilmente ricevibili
Sistemi commerciali[1]	Relativamente alti, soprattutto per l'impianto radio (antenne)	Buona — rimangono i problemi relativi alle intemperie	Ottima — vengono utilizzati sistemi di modulazione digitale non intercettabili

Tabella 2

[1] La totale mancanza di legislazione sul settore commerciale delle reti wireless (per ora destinate solo all'uso "pubblico" di polizia, carabinieri, ambulanze eccetera) blocca in Italia un mercato che è da tempo attento e sarebbe anche immediatamente recettivo.

Come vediamo uno dei problemi principali del *wireless ad alta velocità* è la vulnerabilità ad alcune condizioni atmosferiche. Si possono combattere le intemperie aumentando la potenza di trasmissione e il diametro delle parabole, ma in questo modo si fanno salire vertiginosamente i costi di installazione dell'impianto. Anche la sicurezza presenta delle incertezze e non deve mai essere sottovalutata quando si decide di affidare i propri dati a un mezzo come l'etere. Qualunque persona è in grado di sintonizzarsi sulla frequenza che trasmette i dati per provare a riceverli e decodificarli. Questo inconveniente si può combattere utilizzando modulazioni digitali (come il GSMK, lo stesso sistema utilizzato dalla telefonia mobile GSM, che però non è così sicuro come viene pubblicizzato); in questo modo si raggiungono standard di sicurezza più alti a scapito però di minori velocità e di un maggior costo dei modem. Naturalmente nulla vieta di utilizzare tecniche di crittografia classica, ma ben sappiamo quale ritardo porterebbero in un sistema che vuol essere ad alta velocità.

Per approfondire l'uso delle microonde nelle trasmissioni digitali è consigliabile reperire in rete materiale e informazioni più dettagliate. Si consiglia di cercare le realizzazioni di *Matiaz Vidmar*, un radioamatore sloveno fra i maggior esperti mondiali in comunicazioni packet ad alta velocità; sul suo lavoro si basa lo stato dell'arte nel campo delle microonde e dei radiomodem. Come base per una buona infarinatura iniziale, basta invece sfogliare l'area wireless di Yahoo, che permette di connettersi con i maggiori sviluppatori di queste tecnologie.

I SISTEMI COMMERCIALI: WIRELESS LAN & SPREAD SPECTRUM

Iniziano a essere presenti in maniera sempre più massiccia sui mercati europei (e anche italiani) numerose proposte di schede di rete che permettono la realizzazione di *wireless lan*. Queste schede non sono altro che normali schede ethernet, utilizzate comunemente nelle normali reti locali, su cui è stato aggiunto un modem radio (di solito con una velocità mai inferiore ai 2 Mbaud) e una antenna radio integrata che permette collegamenti nell'ordine dei 300-400 metri all'interno di un palazzo. Come avrete già capito questa tecnologia permette di collegare varie macchine all'interno di edifici senza dove cablare fisicamente le varie stanze. Dal punto di vista legale non è stata ancora assegnata ufficialmente una frequenza per questo tipo di applicazio-

ne, ma l'utilizzo di basse potenze fa sì che molti impianti siano già stati attivati senza arrecare o ricevere disturbi da altri servizi già allocati sulla frequenza in uso.

Molte sono le applicazioni di queste schede: ad esempio l'immediata connessione in rete di macchine situate in luoghi in cui non è possibile (o non è facile) disporre i cavi per un collegamento fisico. La tecnologia utilizzata per raggiungere velocità così elevate si chiama *spread spectrum* e utilizza un'ampia gamma di banda che viene "spazzolata" ad alta velocità dal ricevitore e dal trasmettitore. In questo modo vengono utilizzati più canali radio assieme dove i dati vengono trasmessi in maniera compressa, così da sfruttare al massimo la larghezza di banda. Il limite di questa tecnologia è rappresentato dal veloce decadimento delle prestazioni quando sono attive numerose stazioni (schede) poiché le frequenze sono condivise fra tutti gli utenti. Non è quindi possibile una condivisione di applicazioni o il trasferimento simultaneo di grandi quantità di dati.

Telematica e reti cellulari

A partire dal 1990 il nostro paese è stato completamente invaso da *celle* utilizzate per la telefonia mobile, prima del sistema analogico TACS e poi della rete digitale GSM che per la prima volta ha visto scendere in campo un gestore privato. Queste celle formano una rete che viene utilizzata prevalentemente per il traffico di *fonia* (il comune traffico telefonico), ma che si può facilmente adattare anche al traffico di dati. Cerchiamo di capire come è possibile utilizzarla in maniera ottimale.

La prima analisi si rivolge alla rete analogica TACS che viene gestita da Telecom Italia Mobile e che copre ormai quasi la totalità del territorio del nostro paese. Se per l'utilizzo come rete fonica la qualità raggiungibile è spesso superiore alla sua sorella GSM, la rete TACS non è mai stata progettata per il traffico di dati. Questo difetto è riscontrabile soprattutto nella gestione dello scambio di celle in caso di segnale in movimento; per la fonia qualche disturbo audio è accettabile ma per uno scambio di dati attraverso il modem crea molti problemi. Una possibile soluzione potrebbe essere quella di effettuare il traffico dati sulla rete TACS *senza muoversi* da una cella a un'altra. È una soluzione accettabile, anche se spesso il cambio di cella avviene lo stesso per problemi di saturazione delle frequenze e potremmo dunque vedere la nostra connessione cadere anche se stiamo immobili. A questo si aggiunge la

scarsa sicurezza della rete TACS, facilmente intercettabile con un radio scanner sintonizzato sui 900 Mhz: a causa di entrambe queste limitazioni non si può che sconsigliare questo tipo di traffico.

Ben altro discorso è applicabile alla rete cellulare GSM, standard pan-europeo gestito sia dalla solita Telecom Italia Mobile sia da Omnitel, società il cui maggior azionista è una vecchia conoscenza degli smanettoni, l'italianissima Olivetti di Ivrea.

Il sistema GSM offre vari tipi di traffico digitale essendo interamente basato sulla digitalizzazione dei segnali. Fra i servizi digitali sono ormai famosi i *messaggi SMS*, pacchetti di 160 caratteri che vengono visualizzati sul display del telefono e che possono essere trasmessi tra cellulari della stessa rete o anche tra reti diverse o addirittura arrivare o essere destinati a Internet.

Gli SMS costituiscono un elemento di altissima potenzialità comunicativa. La velocità in cui vengono propagati sulla rete, la futura possibilità di concatenare più messaggi per aumentare la capacità in caratteri (funzione detta *GSM fase 2*) e soprattutto lo scambio con reti diverse da quella originale cellulare ne fanno uno degli strumenti più interessanti per sfruttare la rete GSM. Ogni cellulare può essere trasformato in un *pager alfanumerico* che ci avverte con un piccolo squillo dell'arrivo del messaggio, senza distruggere (come spesso avviene) la pace di chi ci sta vicino.

Per il traffico dati puro è stato implementato un protocollo di correzione di errore (*MNP10*) che permette di raggiungere velocità di 9600 baud anche in movimento sulla rete stessa. Questa velocità viene utilizzata sia per fax che per traffico dati, permettendo quindi accessi liberi da filo da quasi tutto il territorio.

La rete GSM utilizza un sistema di crittografia dei dati che, a detta dei due gestori, mette chiunque al sicuro da intercettazioni. Su questa affermazione c'è molto da obiettare (vedremo più avanti i motivi). Per ora, basti dire che i ponti di collegamento tra le varie celle (sia via radio sia via filo) sono indifesi.

Di chi è l'etere?

Analizzati i vari sistemi per indirizzare i dati via radio è importante fare luce sull'aspetto legale di questa pratica di comunicazione partendo dall'idea che l'etere, ovunque ci si trovi (fatta eccezione per le acque internazionali), è regolamentato da una precisa legislazione.

La prima via legale per avere spazio nell'etere è quella di diventare *radioamatori*. Questo termine è comparso più volte nei paragrafi precedenti: ora proviamo a dare una definizione più precisa di questa strana tipologia di persona.

Per radioamatore intendiamo un appassionato di radiotecnica e di comunicazioni via etere che, mediante un esame di elettronica indetto due volte all'anno dal ministero delle poste e comunicazioni, è autorizzato all'utilizzo di varie frequenze per effettuare collegamenti con altri esperti come lui. Una volta superato l'esame si può legalmente accedere a un vasto spettro di frequenze, che vanno dalle onde corte fino alle microonde, dove il radioamatore può sperimentare varie tecniche di comunicazione. Tra queste tecniche c'è naturalmente il packet radio nei modi che abbiamo appena presentato e l'unica limitazione (poco seguita) che viene imposta è quella di trattare nelle comunicazioni argomenti esclusivamente legati alla radiotecnica. Per avere maggiori informazioni su questo esame e su come diventare radioamatori ci si può informare nelle sedi ARI (Associazione Italiana Radioamatori) che si trovano in ogni città.

Oltre all'esame, c'è l'obbligo di pagare un canone annuo di poche migliaia di lire. Sembra poca cosa, ma queste limitazioni pongono vari problemi a chi si vuole avvicinare alla trasmissioni packet radio in parte per la difficoltà dell'esame e in parte per la troppa esposizione di singoli personaggi in caso di progetti di gruppo.

La seconda via legale per accedere all'etere è quella di farsi assegnare una frequenza per uso cosiddetto civile o privato direttamente dall'amministrazione postale. Questa procedura è la stessa seguita per esempio dalla Croce Rossa o dai RadioTaxi. Le frequenze assegnate sono quelle delle specchio delle Vhf o delle Uhf con una larghezza di banda equivalente a quella della fonia che ci permetterà basse velocità (9600 - 19200 baud) con il rischio di dover condividere con altri servizi il nostro spazio nell'etere. Non è nemmeno da sottovalutare il problema economico, visto che il ministero affitta queste frequenze per cifre che non sono mai inferiori a qualche milione.

Il quadro legale non è esaltante e dunque si è pressoché obbligati a occupare illegalmente le frequenze necessarie per costruire le proprie reti packet. Cerchiamo allora di dare qualche consiglio per capire come funziona l'EscoPost, la polizia postale incaricata di controllare le frequenze nel territorio italiano.

Ricordiamo subito che trasmettere su una frequenza senza nessun permesso è un reato penale punibile dalla legge nr.156 del 29/3/73 con una pena fra i tre e i sei mesi di reclusione e un'ammenda massima di venti

milioni di lire. In realtà i controlli non sono così frequenti e il packet radio ben si adatta a essere un sistema difficilmente rintracciabile. Vediamo il perché.

Le emissioni packet in Vhf e Uhf permettono di solito a più utenti di condividere un sistema telematico e quindi multiple sono le sorgenti dei segnali radio. Questo rende molto difficile la triangolazione (un sistema utilizzato per trovare con precisione l'emissione di un segnale radio) da parte degli organi di controllo. Inoltre il packet non è quasi mai (a parte le emissioni punto a punto in microonde) una trasmissione continua, ma è formato da piccoli pacchetti di dati della durata di un secondo trasmessi in successione tra varie stazioni che non permettono una veloce ricostruzione della direzione del segnale. L'uso di tecniche di crittazione dei dati e magari la variazione programmata dei siti di emissione permette di stare al sicuro da controlli e successivi sequestri.

Il problema principale rimane comunque l'affollamento delle frequenze; questo problema è risolvibile mediante uno studio accurato dello spettro in cui vogliamo operare, magari mediante uno scanner collegato a un computer con uno dei vari programmi di analisi disponibili in rete, cercando frequenze libere ai margini delle occupazioni di banda. Il consiglio migliore rimane comunque quello di ascoltare molto le frequenze su cui si vuole iniziare a trasmettere e di cambiarle molto spesso (i trasmettitori più moderni possono essere interamente controllati dal PC via seriale permettendo di fare cambi di frequenza completamente automatici) riducendo così al minimo l'occupazione della banda in modo da non dare molto fastidio ad altri servizi già allocati. È da notare infatti che il più delle volte le operazioni repressive dell'EscoPost vengono avviate da segnalazioni ben precise da parte degli assegnatari "legittimi" delle frequenze disturbate dai pirati.

La falsa sicurezza delle reti GSM

Quando è stata lanciata sul mercato, la rete GSM è stata offerta come una rete "sicura", in cui l'intercettazione delle telefonate da parte di radioamatori, *phreaker* o magari anche da parte delle istituzioni, era impossibile. Il servizio di sicurezza offerto dalla rete GSM riguardava la possibilità di crittare il collegamento tra la parte mobile della rete (il nostro telefono) e la BTS (la cella a cui siamo collegati). L'algoritmo utilizzato per crittare i pacchetti di dati contenenti la nostra voce (il GSM utilizza la modulazione digi-

tale GSMK e un algoritmo TDMA per l'accesso simultaneo di più stazioni al canale radio) viene indicato con la sigla A5. Non essendo un prodotto di pubblico dominio, i suoi sorgenti di sviluppo non sono liberamente reperibili. Già solo questo dato dovrebbe indurci a una maggiore prudenza e a legittimi dubbi sull'affidabilità dell'algoritmo stesso.

La nascita dell'A5 non è espressamente legata alla rete GSM. L'algoritmo venne adottato dopo una lunga battaglia tra i vari paesi appartenenti al consorzio paneuropeo da cui nacque la necessità di creare un sistema di telefonia digitale sicuro e standardizzato. Già a metà degli anni '80 era viva la discussione sul tipo di algoritmo di crittografia da utilizzare. Paesi come la Francia fecero richiesta di adottare un sistema *non completamente sicuro* e vari servizi di intelligence di altri paesi Nato europei misero in discussione la reale utilità di un sistema cellulare *non intercettabile* neanche dalle autorità. Altri paesi invece si dimostrarono contrari all'attivazione di un sistema implicitamente violabile. La Germania per esempio, al contrario della Francia, avendo subìto traffici di spie con intercettazioni incredibili sul proprio territorio diviso fra est e ovest, sarebbe stata ben contenta di implementare un sistema praticamente inviolabile.

Nella competizione vinse tuttavia la linea morbida e per la rete GSM venne adottata una versione modificata dell'A5, chiamata AX. Questo sistema si basa su un codificatore di tipo *stream cipher* (utilizzato per cifrare un ciclo continuo di bit da trasmettere) che utilizza secondo i parametri ufficiali ETSI una chiave di cifratura di 64 bit per codificare i 114 bit di ogni burst (pacchetto di dati) che viene trasmesso da base a unità mobile.

Secondo l'esperto di crittografia inglese Ross Anderson l'ultimo dato non è reale e la chiave effettiva sarebbe di soli 5 byte (40 bit). Ciò rende fattibile la ricerca della chiave di partenza avvalendosi di soli pochi giorni di calcolo di una workstation o di un potente personal computer. Secondo Anderson sarebbe molto facile realizzare dei chip progettati apposta per realizzare un banale attacco a 2^40 combinazioni, che permetterebbero la nascita di un fantastico cracker di A5.

Nel giugno del 1994 una possibile implementazione dell'A5X è stata diffusa su Internet e vari sono i gruppi che stanno lavorando per la realizzazione di un veloce e funzionale generatore di chiavi. Sempre nel giugno 1994 si doveva tenere a Londra un incontro organizzato dal professor Simon Sherpherd della Bradford University sui problemi di sicurezza degli algoritmi stream cipher e soprattutto dell'A5. Incredibilmente il GCHQ (il servizio inglese di intelligence) riuscì a far saltare la presentazione.

Un ulteriore segnale della effettiva non sicurezza di questo sistema lo possiamo apprendere da un documento scritto da Marcello Scatà e Lorenza Romano (disponibile liberamente in rete) della facoltà di Ingegneria Informatica di Padova, su cui troviamo il seguente passaggio:

"Supponiamo che effettuare una ricerca esaustiva di tutte le possibili chiavi sia il metodo più efficiente per decifrare un messaggio cifrato. Possiamo allora considerare la dimensione della chiave come una misura dell'affidabilità di un algoritmo di crittografazione. Se assumiamo una *cracking machine* capace di un milione di crittografazioni al secondo, otteniamo i seguenti risultati:

Dimensione chiave in bit	32	40	56	64	128
Tempo richiesto per verificare tutte le possibili chiavi	1,19 ore	12,7 giorni	2.291 anni	584.542 anni	10,8*10^24 anni

Possiamo considerare in alternativa il numero di macchine che sarebbero necessarie per decifrare il messaggio in un determinato periodo di tempo.

Dimensione chiave in bit	1 giorno	1 settimana	1 anno
40	13	2	-
56	836.788	119.132	2.291
64	2,14*10^8	3,04*10^6	584.542
128	3,9*10^27	5,6*10^26	10,8*10^24

Nel valutare l'affidabilità di un algoritmo di crittografazione deve essere perciò considerata la "durata" delle informazioni che devono essere protette. Assumendo ad esempio che l'algoritmo A5 utilizzato nel sistema GSM abbia, come sembra, una effettiva chiave di 40 bit (e non 64 bit), fornisce una adeguata protezione per informazioni che hanno un tempo di vita breve. È opinione comune che le conversazioni telefoniche cellulari abbiano un tempo di vita utile dell'ordine di qualche settimana".

Sempre citando Anderson, il lavoro per creare un A5 cracker potrà essere lungo, ma la possibilità di ascoltare le conversazioni della Royal Family e diffonderle liberamente ci dovrebbe mettere tutti al lavoro…

Ed è stato proprio così, via radio, che Lex Luthor mi ha detto che il colpo ormai era fatto. Amico, il sindaco aveva finalmente mollato il miliardone.

Con le lacrime agli occhi, chiaro, ma ormai l'avevano capito che gli conveniva. E poi ho idea che ai pezzi grossi che stanno dietro la Federal Bank di Metropolis mica gli sfagiolava troppo che la faccenda andasse ancora per le lunghe, dopo tutto quel casino che aveva combinato l'Ometto d'Acciaio. Sta di fatto che Luthor mi chiama, tutto giulivo: istruzioni. Rivelare al capo della polizia dove stava la bomba. Poi sparire, e rifarmi vivo con lui quando le acque fossero più calme. No, amico, non mi chiedere come ha fatto il vecchio Lex a incassare il contante senza farsi fregare, non lo so. Ma lui è uno con molte risorse, te l'ho detto. Comunque la scena è stata comica, credimi. Quando polizia, pompieri, FBI, esercito, tecnici delle fogne, cervelloni atomici e tutta la compagnia dei salvatori della città è arrivata lì, c'ha trovato pure Lois Lane e Jimmy Olsen ancora ammanettati alla Bomba. Più morti che vivi dopo tutto quel tempo, ma pur sempre ancora vivi. Un po' palliducci, questo è vero, ma lei è una dura a modo suo, te l'ho detto, e non mi stupisce mica che ce l'abbiano fatta. Al e Louie sono più teneri di quel che sembrano, per cui li avevano legati, sì, ma senza tirare troppo la corda, non so se mi spiego. Quel che bastava perchè potessero bere dal rigagnolo che gli passava tra le gambe, a quanto pare. Beh, non è che fosse proprio acqua di sorgente quella, ma per non crepare di sete gli è bastato. Poi c'erano i topi. In effetti i nostri giornalisti qualche dito del piede se lo sono ritrovato un po' più corto di prima. Ma sai, se un topo viene lì a rosicchiarti l'alluce, con un po' di fortuna puoi anche riuscire a schiacciargli la testa con il calcagno. E qualcosa da mangiare la rimedi. Acqua di fogna e topo crudo. Hai ragione, amico, come dieta un po' di diarrea la fa venire. Per cui penso che per la cara Lois trovarsi senza le mutande in fin dei conti sia stato meglio. Comunque sta di fatto che quando li hanno portati fuori dalle fogne, il nostro amico Superman era lì ad aspettarli. Era un po' mogio, ma quando ha visto la sua donna si è illuminato. Ha messo su il suo faccione da SuperEroe e le ha detto fiero: "Se non altro ti ho riportato le mutandine, cara". E ha fatto un passo avanti tutto impettito, come se si aspettasse un bacio. Beh, amico, a buttare asteroidi nel sole e a

sfasciare mezzo mondo il nostro SuperMuscolo è un asso, ma di psicologia femminile non ne ha mai capito molto. Perchè Lois gli ha fatto un gran sorriso, e per un attimo è sembrata di nuovo la bella pupa di una volta, mica la rovina puzzolente tirata fuori da una fogna che era in quel momento. E con quel gran sorriso in faccia gli si è avvicinata pure lei e gli si è strofinata contro. Gli ha messo una mano tra le cosce, come per una carezza di quelle a 50.000 volts, e poi, sempre sorridendo, gli ha strizzato le palle a tutta forza. Beh, uno si aspetterebbe che l'Uomo d'Acciaio sia fatto d'acciaio pure da quelle parti, no? Ma a giudicare dallo strillo che ha cacciato Superman, ti assicuro che non è così. Poi si è scostata, sempre con quel sorriso in faccia, e senza dire una parola se n'è andata mano nella mano con Jimmy Olsen. Cosa sia successo poi a quei due non lo so di preciso, ma so che Perry White quando invece del servizio in esclusiva si è visto arrivare due lettere di dimissioni per poco non si strozzava col suo sigaro. In città si dice che ora quei due vivono assieme in campagna. Nel Vermont, pare. Lui coltiva petunie. E lei sta scrivendo un libro che dovrebbe chiamarsi "Amore nelle fogne".

Beh amico, da tutta questa storia il povero Superman non si è più ripreso. "Gravissima sindrome depressiva con rischio di esplosioni maniacali compulsive" la chiamano. Sta lì fermo senza dire nulla. Guarda nel vuoto e se gli parli non risponde. Solo ogni tanto fa qualche verso privo di senso: "Gna, gna, gna, piggippì, piggippì", per esempio. Oppure tira fuori di tasca un paio di mutandine da donna e se le rigira tra le mani per ore e ore. Per adesso se ne sta nella sua cella di cemento con muri spessi sette metri e nessuno sa di preciso cosa farne. La DC Comics aveva chiesto al governo di poter mettere una telecamera nella cella, per i bambini che volevano vedere che effetto faceva Superman nella sua camicia di forza al titanio. Così almeno con 50 cents a ingresso recuperavano un po' di spese. Ma lo Zio Sam non ne ha voluto sapere: "Troppo pericoloso, non si sa mai" hanno detto...

A quel punto era passato un po' di tempo e le acque mi parevano calme abbastanza, non so se mi spiego. E mi sono fatto vivo con Luthor. Il vecchio serpente era di ottimo umore. Quando sono entrato nel suo ufficio, là in cima al grattacielo della LL Corporation, sulla scrivania di palissandro c'era una bottiglia di champagne che mi aspettava. Dentro un secchiello da ghiaccio d'argento e con due coppe di cristallo bordate in oro. E' uno che ha classe il vecchio Lex, non c'è che dire. "Al meritato successo della nostra impresa" mi fa, riempiendomi il bicchiere. Bah, lo champagne a me sembra gazzosa, ma quando te lo versa lui non puoi mica stare a dirgli "Grazie Lex, ma preferirei un po' di bourbon", no? E poi attacca a sviolinarmi per un po'. Siamo grandi amici noi due, anzi sono il suo figlio prediletto, tutto questo un giorno sarà mio e bla, bla, bla. Finita la sviolinata, mi guarda dritto in faccia. Sorride sempre, ma gli occhi sono come quelli di uno squalo pronto ad azzannare: "Naturalmente abbiamo ancora un piccolo da problema da sistemare, vero?". "Temo di sì, capo" gli dico io, che so già dove vuole arrivare. "Vi avevo ordinato di non muovervi per nessuna ragione, non è stata una buona idea uscire dal nascondiglio quella sera e lasciare che quei giornalisti rischiassero di compromettere tutto". "No capo,

non è stata una buona idea" rispondo io. In realtà penso: "Se non uscivamo, Jimmy Olsen e Lois Lane restavano fuori a spiarci invece di entrare, e ci poteva andare molto peggio, capo. E mica ci hanno scovati per colpa nostra, capo. Ci hanno scovati perchè c'è stato un fottuto buco nella TUA fottuta organizzazione, capo". Ma mica glielo dico, questo, al vecchio Lex. "Bene bene" fa lui. "Allora capisci da te che Al e Louie ora sanno troppe cose. Mi sembra giusto che sia tu a provvedere, non trovi?" "Sì capo. Mi sembra giusto". "Bravo ragazzo" seguita lui "quando pensi di poter risolvere il problema?" "Domani notte, capo, se per te è Ok". "Perfetto" fa Lex, tornato tutto paterno "lo so che posso sempre contare su di te, Joe. E per dimostrarti la mia stima e il mio affetto, quando tornerai da me dopo esserti occupato di quei due, troverai una bella gratifica di cinquantamila dollaroni. Passerò sopra al tuo errore. Sei contento?" "Sì, capo" faccio io, con tutta la faccia di bronzo che riesco a racimolare. E credimi amico, non è mica poca. "Certo che sono contento. Lavorare al tuo servizio è sempre un piacere, capo". Lex mi congeda con un cenno, mentre preme un pulsante per chiamare la sua nuova segretaria. Quella che ha vinto il concorso di Playboy lo scorso mese. Ma mentre esco dall'ufficio sono incazzato nero, altro che contento. Di Al e Louie me ne potrei anche fottere, capisci? Non muoio di entusiasmo all'idea di farli fuori, ma sono inconvenienti del mestiere. Ma 50.000 dollari? Per un colpo da un miliardo??? Una miseria come quella, dopo che il vecchio coccodrillo se n'è rimasto tutto il tempo ad abbronzarsi il culo alle Bahamas mentre il qui presente il culo rischiava di farselo bruciare da Superman? Eh, no, cazzo! Senza contare che Lex Luthor sopra agli errori non ci passa mica tanto facilmente. Oh, no. Già me lo immagino cosa vuol fare. Io stendo Al e Louie. E domani qualcun altro stende me. E per incassare quella miseria di 50.000 io buono buono torno da Lex come un vitellino dal macellaio. Grazie tante, Lex, grazie davvero.

Ragion per cui amico, mi sono messo subito a pensare ai miei guai. Potevo darmela a gambe anche subito, è vero, ma me l'aveva sempre detto la mia nonna: "Joe, tu lavora con i pezzi grossi invece che con i perdenti, è meglio per te. Ma ricorda sempre che lavori con loro, non per loro. E tieni sempre un asso nella manica, perchè verrà il momento di lavorare per te". E io un asso nella manica con Lex me l'ero tenuto, amico. Vedi, lo so che se ti dico che il vecchio Lex ha dei sentimenti tu mi schiatti dalle risate qua sul tavolo. Eppure io so una cosa di lui che non sa nessuno, amico. Se la lasciò scappare una volta che era pieno di quella gazzosa. A proposito, perchè non ci facciamo un altro bicchiere pure noi? Il fatto è che pure al vecchio Lex è toccato innamorarsi una volta in vita sua. Un'attrice, e se ti dicessi chi era quell'attrice non mi crederesti. Fu quando Lex ancora non era il "vecchio Lex", capisci? E ogni tanto faceva ancora dei lavoretti per il governo. Per cui quando quell'attrice si mise in un pasticcio con un presidente poi morto ammazzato, una gran brutta storia, fu proprio a Lex che toccò metterla a nanna, mi spiego? "Suicidio con sonniferi" 'sto cazzo. Naturalmente il giovane Lex era già un professionista serio, per cui non ci pensò su due volte, amore o non amore. Come professionista tanto di cappello pure a lui, non c'è che dire. Ma mentre mi raccontava quella storia, non ci crederai, quasi piangeva... O forse era la troppa gazzosa che gli usciva dagli occhi, chissà.

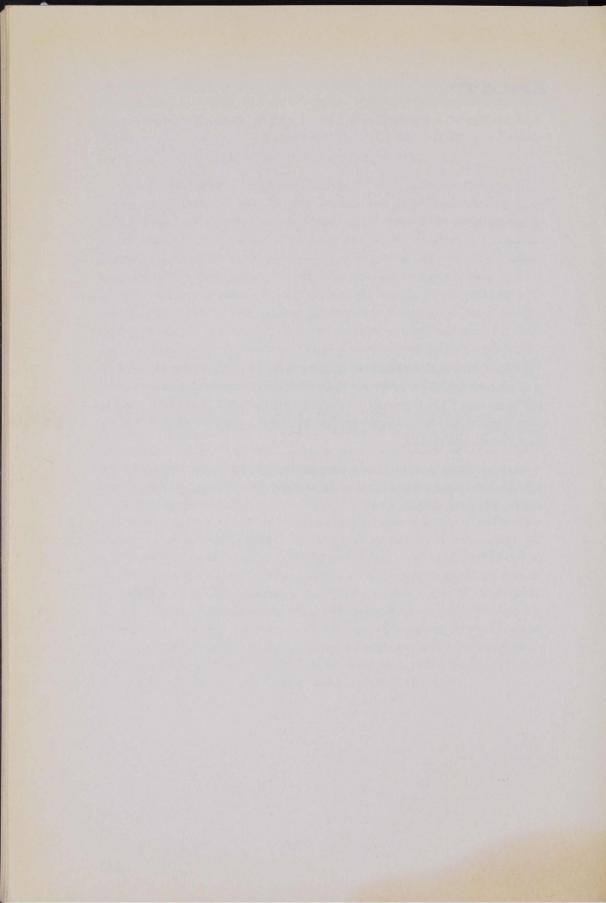
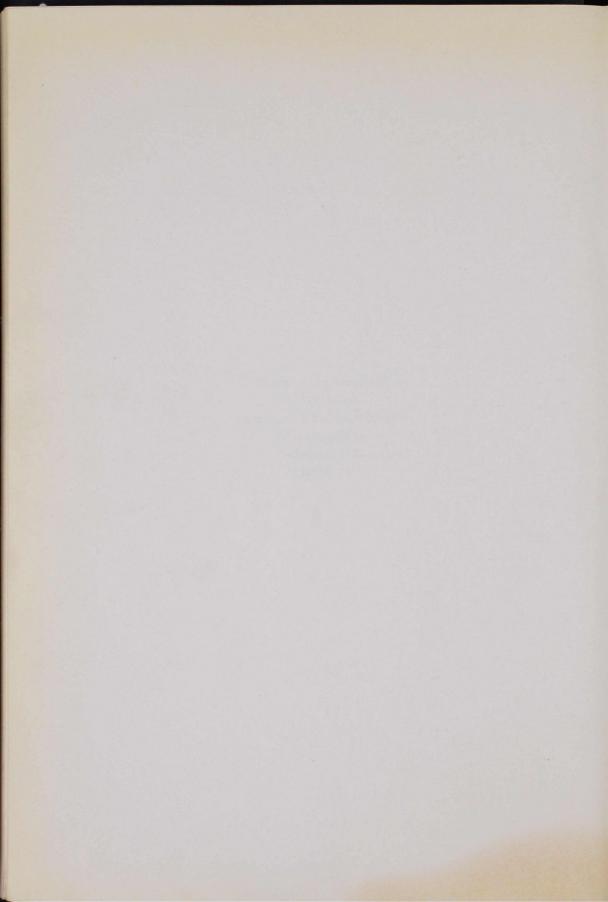
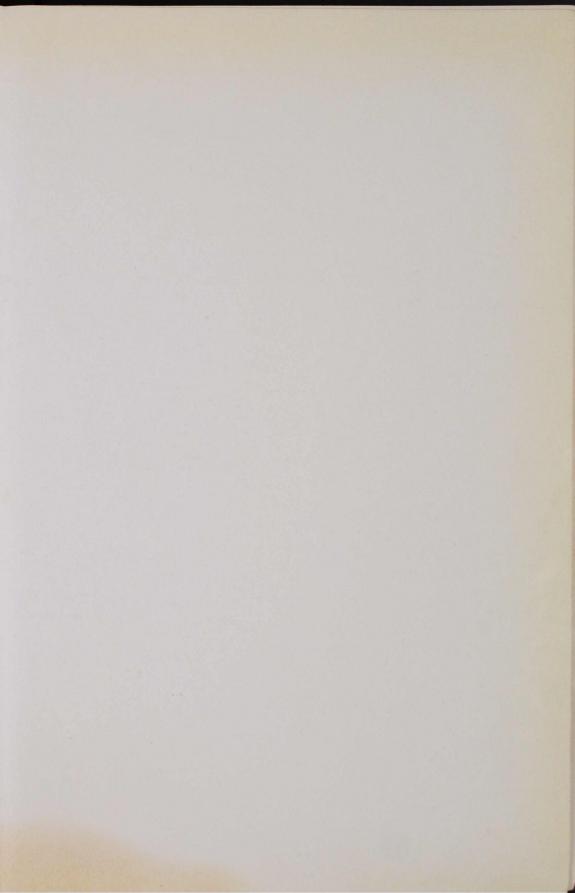
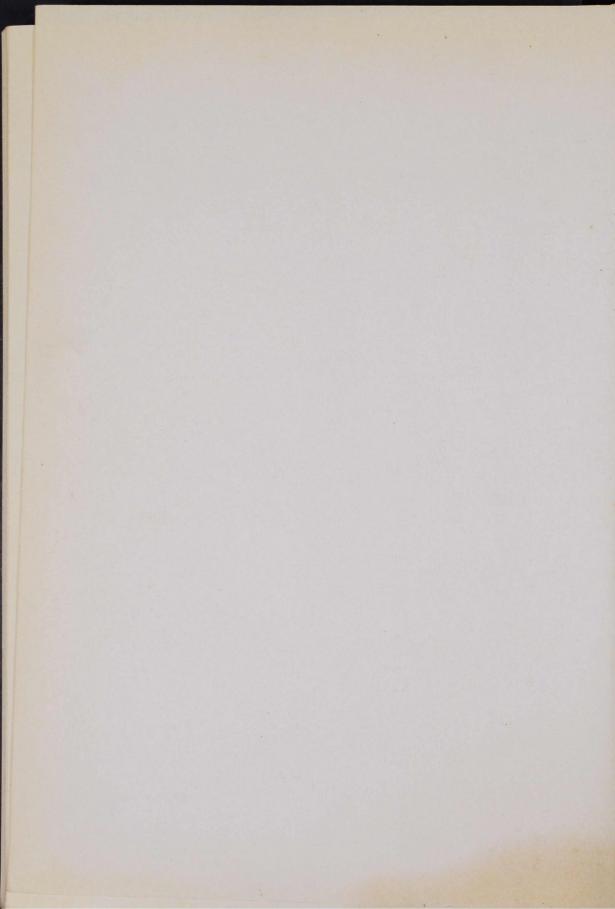
Comunque, un'ideuzza mi era venuta. Il pensiero di Luthor che si godeva il miliardo e del qui presente Joe Lametta in pasto ai vermi mi suggeriva che era arrivato il momento di prendermi la paga che mi spettava per tutti gli anni passati a lavorare con Lex. "Soddisfazione unilaterale dei propri bisogni". Oh, sì. Un altro parolone, ma il vecchio Luthor sarà contento di come ho imparato bene la lezione, amico.

Quando sono entrato di soppiatto nell'ufficio di Lex la notte stessa ho passato la peggiore strizza della mia vita. Ma come ho detto sono un professionista anch'io. E un'idea di come fare a superare tutto lo sbarramento ce l'avevo da un pezzo, almeno da quando avevo iniziato a mandare a memoria dove stavano le guardie e i sistemi di allarme del palazzo. Il computer personale di Luthor stava lì, sulla scrivania. Nessuna particolare difesa, ci avrei giurato. Il vecchio Lex non ha mica voglia di stare a perdere tempo, lui. Password di sistema, filesystem crittati, non era roba per lui quella, tanto nel suo ufficio mica poteva entrarci qualcuno, pensava. Beh, si sbagliava di grosso, stavolta. Solo un po' di roba passata al PGP nel suo computer, ci avevo scommesso e ci avevo azzeccato. Ma la scommessa vera veniva adesso. Cercavo un'informazione molto precisa, e quell'informazione non poteva che trovarsi proprio all'interno di quel file crittato. Conoscevo Luthor da troppo tempo ormai, e sapevo quello che stavo facendo.

Provo la prima password: "Marilyn". Nulla. "Error: Bad pass phrase." Merda. Possibile che avessi fatto male i miei calcoli? Provo con la data della morte. Nulla. Provo con la data di nascita: "7.1.1926". Tombola amico! Come quando vai a Las Vegas e vinci tutto il pot al primo colpo. E naturalmente c'era tutto là. Il vecchio Lex su 'sta faccenda della crittografia aveva perso un po' di colpi, o forse era stata la povera Marilyn a rimanergli troppo nel cuore, chi lo sa. Ma a parte quello era stato furbo come sempre. Chi sarebbe andato a pensare che in uno scassato armadietto bagagli a combinazione alla stazione della GreyHound c'era una valigia con un miliardo di dollari dentro, eh? Già. Proprio una vecchia sudicia valigia come questa. Ehi, che ti succede ora? Ti vedo un po' verde in faccia. Sarà stato tutto quel whisky, amico. Non preoccuparti, ora ti fai un bel sonnellino su quel divano, mentre il qui presente ti saluta e se ne va all'aereoporto. Lasciala perdere la pozza di vomito, ora ti accompagno al divano. Meglio per tutti se ti fai un bel sonno e ti svegli domattina, uomo. Nessuna preoccupazione per me. E nessuna per te. Avrai un bel mal di testa domani, amico, ma anche una bella storia da raccontare ai tuoi nipoti fra qualche tempo. Ricorda questo nome: Joe Lametta. Il whisky che ti sei bevuto te lo paga tutto lui. Sono in grana ormai, te l'avevo detto, no?

Epilogo

Di Joe Lametta, del miliardo di dollari e della sua carriera criminale — che ha compreso anche un breve ma indimenticabile periodo di presenza sulla rete telematica FTN Cybernet — dopo quella notte si è persa ogni traccia, salvo il misterioso testo steganografico riportato a pagina 5 di questo libro (pare che sia una dedica).

Non sappiamo dove si trovi e cosa stia facendo in questo momento, ma gli auguriamo buona fortuna. E testimoniamo che tutto quello che ha raccontato è la pura verità.

Dai bassifondi di Metropolis, tra cantine malfamate, topi e spazzatura, Joe Lametta intraprende la strada del crimine fino a diventare il luogotenente di Lex Luthor, terrore e incubo della città.
All'apice della sua carriera un'operazione rocambolesca gli permette di rubare un miliardo di dollari truffando il sindaco, la polizia, Superman e infine il suo stesso padrone. Le sue armi sono un modem e un computer portatile; un manuale lo aiuta spiegando le tecniche per comunicare nelle situazioni più difficili conservando anonimato e sicurezza.
Di Joe Lametta e del miliardo di dollari si sono perse le tracce, ma ci è rimasto il manuale e il racconto della sua impresa.
Crittografia, PGP, file system crittati, anonymous remailer, nym server, steganografia, telefonia digitale crittata e comunicazione via etere. Più un'ampia introduzione che spiega perché, nella società del controllo, le reti digitali possono costituire il luogo più sicuro in cui condurre i propri affari.
Frutto di ricerche e prove meticolose, questo libro vuole essere un fidato maestro d'armi per tutti coloro che non si accontentano delle luci colorate di Internet.
Le istruzioni chiare e complete, gli esempi e il linguaggio preciso ma accessibile a chiunque ne fanno uno strumento dalle conseguenze imprevedibili: i suoi lettori potranno essere impiegati e top manager, mafiosi e amanti discreti, trafficanti di droga e militanti per i diritti civili. Ognuno saprà trovare la sua strada per mettere in pratica quanto vi è descritto.
Unici elementi in comune: l'amore per l'individuo e per i suoi bisogni, il disprezzo per una società poliziesca che si vuole prendere troppa cura dei suoi cittadini.

L. 23.000